Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Triennale in
Ingegneria Informatica

Caratterizzazione e studio di una sorgente di fotoni entangled

Relatore:
Prof. GIUSEPPE VALLONE

Correlatore:
Dott. GIULIO FOLETTO

Laureando:
MICHELE RIEPPI

N° matricola:
1218669

Anno Accademico 2021/2022

Abstract

Una sorgente di coppie di fotoni entangled è un dispositivo complesso le cui prestazioni dipendono da numerosi parametri. Una caratterizzazione dettagliata è necessaria per operare correttamente questo dispositivo.
In questo lavoro di tesi, tramite degli script sviluppati in linguaggio python, è stato possibile mettere in relazione i valori di corrente e temperatura del laser di pompa, con il suo spettro in lunghezza d'onda, misurato dallo spettrometro di laboratorio. Questi script permettono di comunicare (tramite porta seriale) e controllare automaticamente i dispositivi che compongono la sorgente, e quindi aiutano a verificare direttamente e in maniera più ripetibile le condizioni fisiche migliori in cui essa esprime le sue potenzialità. Così facendo, si può aumentare il numero di coppie prodotte e migliorare la qualità dell'entanglement.
I fotoni prodotti possono poi essere utilizzati poi per scopi di laboratorio e di ricerca.

Indice

Capitolo 1

Introduzione

La fisica quantistica permette la descrizione di quanti di materia, l'informatica permette la rappresentazione dell'informazione tramite bit: l'unione di queste due scienze permette di rappresentare, raccogliere e analizzare dati con l'uso del qubit. Il qubit è l'elemento base della quantum information, e a differenza del bit classico, che può avere solo due valori (0 o 1), può essere descritto da una sovrapposizione dei due. Questo permette un aumento esponenziale della potenza di calcolo per alcuni problemi.
Un fotone [1] è un quanto di luce, ed è ciò che permette di rappresentare un qubit. Infatti, la sua polarizzazione può essere descritta come un sistema quantistico a due livelli. Inoltre, tramite dei particolari dispositivi è possibile generare stati non classici, come ad esempio fotoni entangled [3], che sono poi la base di diversi studi teorici e applicativi di quantum information.
In questa tesi si mira, dopo una spiegazione teorica di alcuni concetti di fisica quantistica, allo studio dei componenti di una sorgente di fotoni entangled. In particolare è stato sviluppato un software in grado di poter impartire dei comandi ai dispositivi e di salvare i risultati prodotti da questi. E' stata eseguita poi un'analisi dei dati prodotti al fine di poter ottimizzare la sorgente e garantire le prestazioni migliori. I parametri dei dispositivi con maggior successo nella generazione di fotoni entangled, verranno successivamente impiegati per ulteriori applicazioni all'interno del laboratorio di ricerca.
L'affiancamento delle applicazioni informatiche conosciute allo studio teorico maturato da fonti come Quantum Physics di Michel Le Bellac [2] e svariate tesi, ha permesso un approfondimento su un tema non coperto dal corso di studi di ingegneria informatica. Questo ha consentito all'autore una basilare comprensione dell'informazione quantistica, un campo che promette grandi sviluppi nei prossimi

decenni. Tuttavia questo elaborato non ha lo scopo di spiegare argomenti della meccanica quantistica, ma esemplificare come questa possa trovare applicazioni anche all'informatica.

Capitolo 2

Teoria

La fisica classica rappresenta un punto di partenza fondamentale per spiegare i fenomeni della realtà che ci circonda, le sue leggi e le sue peculiarità, ma per alcuni elementi (specialmente quelli più piccoli) la meccanica quantistica prende il sopravvento; questa si occupa di descrivere il comportamento delle particelle piccole, come fotoni, protoni, neutroni ed elettroni. Nell'ultimo decennio l'avvicinamento della quantistica all'informatica ha portato a numerosi studi e ricerche che possono superare a livello computazionale il modello informatico classico basato sulla logica binaria e sviluppato negli ultimi decenni. I sistemi quantistici sono un modo promettente per l'elaborazione dell'informazione e hanno tutte le caratteristiche per diventare una parte fondamentale dell'informatica moderna.

2.1 Qubit e Entanglement

Il bit è l'unità base dell'informazione in informatica e nelle comunicazioni digitali. Attraverso la semplicità dei suoi valori (0 e 1) i dispositivi odierni sono in grado di processare dati e programmi, oltre a comunicare. Esistono tuttavia sistemi non classici, che non possono essere rappresentati tramite soli due valori.
In un sistema quantistico, un fotone, può trovarsi in una sovrapposizione dei due stati. Il qubit, permette di rappresentare uno stato come combinazione lineare dei due valori 0 e 1. Formalmente:

$$|\phi\rangle = \lambda|0\rangle + \mu|1\rangle$$

Nel momento in cui si esegue una misura sul sistema che rappresenta il qubit, lo si proietta su una base di vettori come quella formata da $|0\rangle$ e $|1\rangle$. La probabilità di ottenere valore 0 o 1 è dato da $|\lambda|^2$ o $|\mu|^2$.

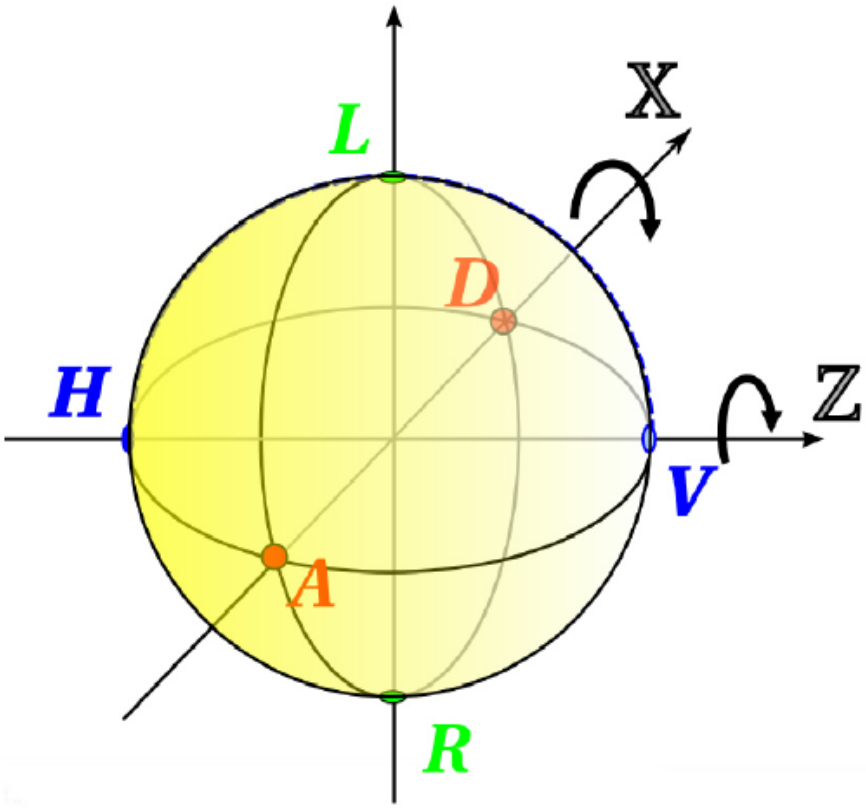

Figura 2.1: Sfera di Bloch rappresentante gli stati puri (o vettori puri)

Una rappresentazione grafica di un qubit o di uno stato quantico a due livelli è data dalla Sfera di Bloch, una sfera di raggio unitario i cui punti sulla superficie sono in corrispondenza con gli stati puri del sistema. Due angoli reali θ e ψ sono sufficienti per rappresentare uno stato puro di un qubit: $|\phi\rangle = cos(\theta)|0\rangle + e^{i\psi} sin(\theta)|1\rangle$.
La polarizzazione dei fotoni, oggetto di studio di questa tesi, è un esempio calzante della teoria del qubit. La polarizzazione di un fotone può essere orizzontale o verticale, $|H\rangle$ o $|V\rangle$, normali alla direzione in cui può oscillare il campo elettrico. Esistono poi degli stati dati dal principio di sovrapposizione, come somma dei due visti precedentemente, questi sono $|D\rangle$ o $|A\rangle$ rispettivamente diagonale e antidiagonale, ossia i vettori ruotati di $\pi/4$ rispetto ad $|H\rangle$ e $|V\rangle$. Nella Sfera di Bloch $|H\rangle$, $|V\rangle$, $|A\rangle$ e $|D\rangle$ sono ortogonali tra di loro. Tuttavia la polarizzazione non è obbligatoriamente lineare, generalmente segue un andamento ellittico o circolare e a seguito del calcolo dei vettori, i risultanti saranno $|L\rangle$ e $|R\rangle$, rispettivamente ortogonali nella sfera di Bloch a quelli lineari.
Il fenomeno dell'entanglement è un fenomeno quantistico, non riconducibile alla meccanica classica in cui due sistemi non sono descrivibili singolarmente.In parti-

colare, le correlazioni fra i risultati delle misure operate sui due sistemi non hanno alcun analogo classico. Secondo la meccanica quantistica è possibile realizzare un sistema costituito da due (o più) particelle che possono mantenere una correlazione delle proprie caratteristiche, facendole interagire o acquisendole secondo un processo che le origini nel medesimo istante, in modo che siano descritte da uno stato quantico globale definito, pur mantenendo uno stato singolare indefinito fino al momento dell'esecuzione della misura. Il processo di misura relativo ad una singola particella è soggetto alle regole quantistiche della probabilità. [6] Studiando il fenomeno dell'entanglement sui fotoni ci permette di stabilire una correlazione tra le polarizzazioni delle coppie generate. Formalmente, un sistema quantistico bipartito è descritto dal prodotto tensore:

$$|\lambda\rangle_A \otimes |\mu\rangle_B$$

In cui i due sistemi A e B sono rispettivamente nello stato λ e μ. Lo stato del sistema composto è dato dal prodotto tensore, ed è separabile nei suoi due fattori. Tuttavia, nel caso più generale, uno stato non è un semplice prodotto tensore, ma una combinazione lineare di più prodotti tensori:

$$\sum_{i,j} c_{ij}|i\rangle_A \otimes |j\rangle_B$$

questo stato non separabile è chiamato stato entangled.
Nell'età moderna questo potenziale può essere utilizzato per varie applicazioni in scopo scientifico [7]. Un' importante applicazione, relativa all'informatica e in continuo studio e sviluppo, è legata alla crittografia: la condivisione di uno stato entangled permette a due utenti di generare chiavi crittografiche simmetriche e sicure. Nel campo scientifico-informatico possiamo trovare altre applicazioni relative all'uso dei fotoni entangled, per esempio nella generazione dei numeri casuali [8].
Il modello non classico si affida alla creazione di fotoni entangled tramite un processo ben preciso, che comprende la messa in gioco di diversi campi della matematica e della fisica, analizzato nel dettaglio nella sezione successiva.

2.2 SPDC

L'SPDC, acronimo di Spontaneous parametric down conversion, è un processo secondo il quale è possibile produrre, coppie di fotoni. Esse manifestano correlazioni nei loro gradi di libertà, come la polarizzazione, e possono essere entangled

se queste correlazioni vengono prodotte in sovrapposizione [9]. Il processo di creazione consiste in una particolare interazione tra un fascio luminoso (dato da un laser di pompa) e un cristallo non lineare.

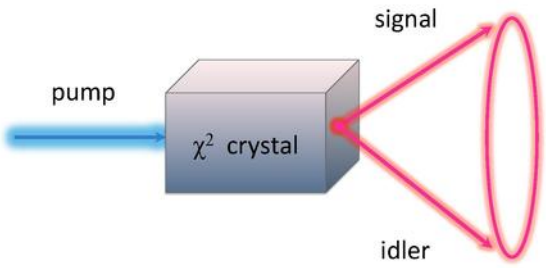

Figura 2.2: Processo SPDC

Dopo l'emissione del laser, la luce viene indirizzata verso il cristallo, il quale grazie a sue particolari proprietà date dalla non linearità del materiale, permette di generare una coppia di fotoni con proprietà fisiche comuni. Alla collisione con il cristallo, questo emette istantaneamente una coppia di fotoni, stimolate dalle fluttuazioni casuali nel vuoto del laser. Le coppie di fotoni sono generate ad intervalli non prevedibili poiché il processo è probabilistico e tipicamente molto raro, per cui solo pochi fotoni di fotoni di pompa vengono convertiti [10].
E' da precisare come, a causa dei principi di conservazione, la somma delle energie e quella delle quantità di moto dei fotoni generati nel cristallo, coincidano con l'energia e quantità di moto possedute dal fotone emesso in principio dal laser.
I cristalli usati per SPDC sono spesso birifrangenti, per cui la polarizzazione ha un ruolo molto importante nel processo. Il comportamento della polarizzazione permette di dividere l'SPDC in due tipi:

- Tipo 1: I fotoni prodotti hanno la stessa polarizzazione che risulta ortogonale a quella del laser di pompa
- Tipo 2: I fotoni prodotti hanno polarizzazione ortogonale tra loro, uno dei quali (signal), mantiene quella prodotta dal laser di pompa.

A seconda del tipo, si può produrre entanglement in polarizzazione in diverse maniere.

Ad esempio, per il tipo 1, si possono usare due cristalli sottili di BiB_3O_6 giustapposti e allineati con i rispettivi assi di birifrangenza ortogonali. Così facendo, il primo cristallo interagisce con la componente (ad esempio) polarizzata verticale della luce di pompa per produrre fotoni polarizzati orizzontalmente, mentre il secondo, viceversa, interagisce con la componente orizzontale e produce fotoni verticali. Con un preciso allineamento, si può perdere l'informazione di quale cristallo abbia prodotto le coppie, rendendo entangled lo stato della polarizzazione. Se i cristalli sono di diverso spessore, è più probabile produrre coppie con quello di dimensione maggiore, e l'entanglement peggiora.
Nel caso sperimentale studiato in questa tesi, si usa invece il tipo 2 e un'altra tecnica di generazione, che verrà spiegata nel prossimo capitolo

Capitolo 3

Struttura della sorgente

L'alternativa adottata al laboratorio di Ingegneria dell'Informazione di Padova per la produzione di fotoni entangled consiste, sempre tramite il processo SPDC (tipo 2), nell'uso di un singolo cristallo non lineare posto al centro di un interferometro di Sagnac [12]. In questa sezione verrà approfondita la struttura sulla quale verranno poi eseguiti i test per la generazione di fotoni entangled [13].

3.1 Architettura fisica

La sorgente è basata su un interferometro di Sagnac polarizzato. Nella figura 3.1 è presente la rappresentazione dello schema della sorgente, nella 3.2 è riportata la realizzazione sperimentale.

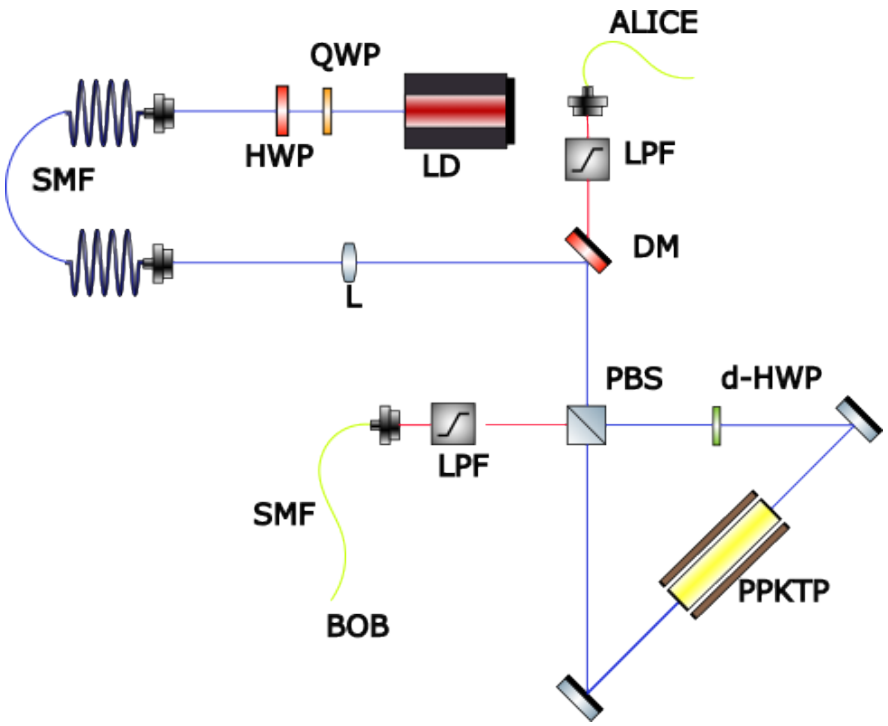

Figura 3.1: Schema sperimentale della sorgente di fotoni entangled basata su un interferometro di Sagnac in polarizzazione. Il laser di pompa è spedito in una fibra ottica che mantiene la polarizzazione della sorgente, successivamente, questa è collegata ad una half-wave plate che ruota la polarizzazione del fascio luminoso di 45°. Al centro dell'interferometro è posizionato un cristallo di potassio titanil fosfato, il laser procede poi verso un polarized beam splitter, posto all'entrata dell'interferometro di Sagnac, che separa i fotoni in una sovrapposizione oraria (V) e antioraria (H). Il fascio orario percorre all'interno di una half-wave plate di lugnghezza d'onda doppia, che cambia lo stato da V a H producendo coppie $|\mathrm{H}\rangle_S\ |\mathrm{V}\rangle_I$, allo stesso modo quello antiorario produce coppie$|\mathrm{V}\rangle_S\ |\mathrm{H}\rangle_I$ (solo dopo che entrambi i fasci sono passati nel cristallo con polarizzazione H). Al ritorno nel PBS i fotoni sono combinati e lo stato risultate è $|\mathrm{H}\rangle_S\ |\mathrm{V}\rangle_I + e^{i\theta}|\mathrm{V}\rangle_S\ |\mathrm{H}\rangle_I$, dove θ è una fase data dalla diversa lunghezza dei due percorsi dell'interferometro. Il passaggio di ritorno alla pompa è rimosso tramite un filtro passa alto (LPF) prima dell'ingresso in una fibra mono-modale.

Nella struttura sono presenti diversi componenti che compongono un'architettura complessa in grado di generare coppie di fotoni entangled. Il laser di pompa (LD) è il componente che genera il fascio luminoso e che, nel nostro caso, ha una lunghezza d'onda di 405 nm. Il laser viene portato subito alla half-wave plate (HWP) e la quarter-wave plate (QWP) permettono il cambiamento della polarizzazione e dello stato del laser catturato poi da una fibra ottica mono-modale. Questa porta il fascio nell'area del banco ottico dove si trova il resto della sorgente e dove poi raggunge l'interferometro di Sagnac, alla cui entrata è presente un polarized beam splitter (PBS) che consente la creazione di una sovrapposizione dello stato orario e antiorario, facendo procedere la polarizzazione V nel primo percorso e H nel secondo. Lo stato è poi cambiato da una dual-wavelength half-wave plate (d-HWP) che cambia la polarizzazione dello stato orario (da V ad H). Entrambi finiscono poi al centro dell'interferometro dove il cristallo potassio

titanil fosfato produce lo stato $|\mathrm{H}\rangle_S\ |\mathrm{V}\rangle_I + e^{i\theta}|\mathrm{H}\rangle_S\ |\mathrm{V}\rangle_I$, dove la prima coppia è il percorso orario e la seconda è il percorso antiorario. Quest ultimo deve ancora infrangersi sulla d-HWP che cambia lo stato della coppia in $|\mathrm{V}\rangle_S\ |\mathrm{H}\rangle_I$. Lo stato di sovrapposizione è composto quindi, prima del nuovo attraversamento del PBS in $|\mathrm{H}\rangle_S\ |\mathrm{V}\rangle_I + e^{i\theta}|\mathrm{V}\rangle_S\ |\mathrm{H}\rangle_I$, dove θ è una fase che tiene conto di una piccola elliticità della polarizzazione di pompa e di lievi asimmetrie fra i due percorsi dell'interferometro.
Dopo aver passato il PBS, che trasforma nuovamente lo stato è necessario far passare gli output in fibre ottiche mono-modali. I flussi vengono poi misurati da Alice e Bob, che ne determinano le coincidenze.

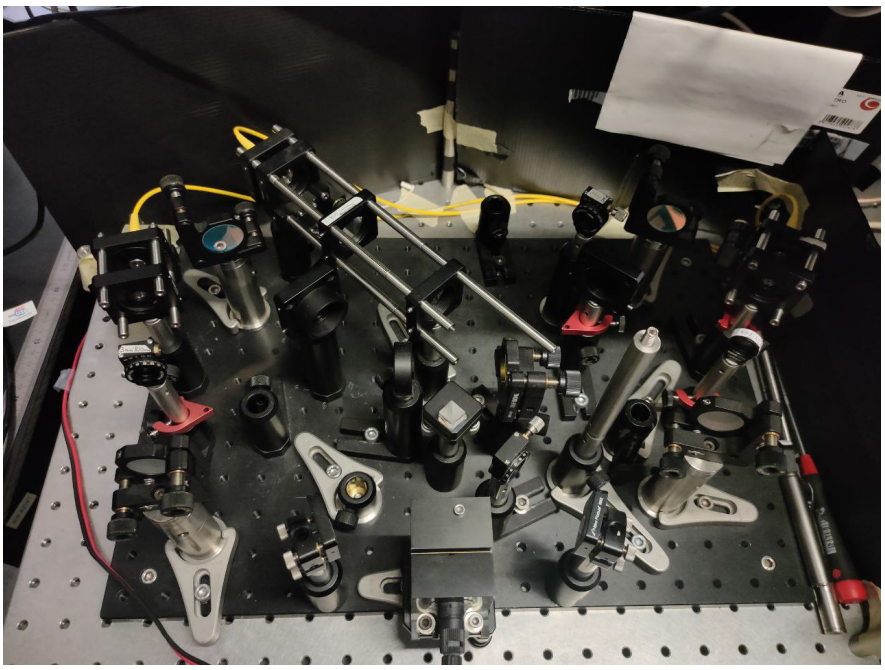

Figura 3.2: Realizzazione sperimentale di laboratorio sul tavolo ottico. La sorgente è stata montata su una breadboard per permettere il corretto posizionamento dei vari componenti della struttura, questi infatti richiedono una particolare precisione per garantire una migliore probabilità di generare coincidenze.

3.2 Laser di pompa

Nel corso della sperimentazione sono stati impiegati due laser con caratteristiche diverse e di conseguenza risultati ottenuti differenti. Entrambi i dispositivi sono stati collegati ad un'alimentazione tramite una presa 12V DC e indirizzati verso la prima fibra ottica che ha il compito di catturare il laser emesso.

3.2.1 Ondax's LM Series Compact Laser Module

Il primo laser con cui è stato possibile generare fotoni entangled all'interno del laboratorio di ricerca è il Ondax's LM Series Compact Laser Module, con una lunghezza d'onda di 405 nm e una potenza di 40 mW. La dimensione del fascio luminoso è di 1.5 x 3.5 mm con una divergenza minore di 10 mrad.

Figura 3.3: Ondax's LM Series Compact Laser Module usato come laser di pompa per l'interferometro di Sagnac. Sorgente [14]

Nel corso degli esperimenti il laser ha operato in un range di temperatura tra i 20 e i 30 °C e un range di corrente tra i 40 mA e i 60 mA, questi due parametri sono stati presi in considerazione in base all'ambiente in cui ci troviamo e alle prestazioni che, all'interno dei range, garantiscono. Temperatura e corrente hanno un'importante influenza sul fascio prodotto, essi possono essere gestiti tramite comandi manuali presenti sull'interfaccia fisica del laser. Come si vedrà successivamente i parametri del laser verranno impostati tramite script per migliorare la generale efficienza del sistema.
Il laser presenta un problema sulla corrente, si è verificato infatti (in fase di manutenzione del software che controllava il laser) che questa viene misurata dall'unità del laser con circa 1.2 mA di difetto rispetto allo script.

3.2.2 Coherent Obis LX

L'ultimo laser messo a disposizione del laboratorio è il Coherent Obis LX, con una lunghezza d'onda sempre di 405 nm, potenza fino a 40 mW, si tratta di un successore tecnologico del modello descritto nella precedente sezione.

Figura 3.4: Coherent Obis LX usato come laser di pompa per l'interferometro di Sagnac. Sorgente [15]

Nel corso degli esperimenti il laser ha operato con parametri di potenza e temperatura equivalenti al precedente, per mettere così a confronto le prestazioni dei due dispositivi e verificare che il nuovo dispositivo potesse sostituire il laser utilizzato prima. La nuova sorgente, inoltre, permette di poter scegliere una operating mode in grado di essere controllato in corrente (mentre la potenza veniva gestita dal sistema) o in potenza (corrente gestita dal sistema); tuttavia non è stato possibile implementare un comando per impostare la corrente (in quanto dalla documentazione questo non era presente) ma solo la potenza erogata.
Tramite il software proprietario è stata possibile un'iniziale comprensione del comportamento del laser, e anche consultando la documentazione fornita, è stato possibile venire a conoscenza della sintassi dei comandi da fornire al dispositivo. E' stato notificato il fatto di non poter controllare la corrente operativa del laser, nonostante la possibilità di leggerla, ma avendo come mezzi di impostazione unicamente la potenza (in percentuale da 0 a 110% o in mW da 0 a 40) e la temperatura. Nel corso delle misure è stato notato che un cambiamento di temperatura superiore a 0.5°C tramite comando seriale portava il laser a spegnersi per impostare il valore passatogli, per poi tornare in stato ready. Essendo questo laser nuovo di fabbrica, non ci sono stati errori particolarmente evidenti o cruciali che possano aver afflitto la strumentazione e il maggior numero di coincidenze create rispetto al precedente.

3.3 Spettrometro

Al fine di poter misurare lo spettro è stato messo a disposizione l'Ocean Optics HR4000. Lo spettrometro permette di visualizzare l'intensità del fascio luminoso

in funzione della lunghezza d'onda a cui questo viene emesso. L'uso dello spettrometro è essenziale per verificare le conseguenze di ogni modifica effettuata sui parametri impostati dei laser sulla qualità dello spettro. E' possibile visionare i risultati real-time ad ogni singolo cambiamento, mentre il software fornisce in output i dettagli dello spettro.

Figura 3.5: Ocean Optics HR4000. Sorgente [16]

Secondo le caratteristiche dei laser, l'output dello spettrometro dovrebbe mostrare un picco di intensità luminosa nella fascia dei 400 nm (quella in cui i laser operano).
Nonostante i dispositivi potessero includere un software in grado di gestire gli stessi, come si vedrà nella sezione successiva, sono stati creati degli script in grado di gestire facilmente gli strumenti protagonisti del processo di generazione di fotoni entangled.

Capitolo 4

Sperimentazione

Il principale lavoro di questa tesi è stato lo sviluppo software per la gestione dei dispositivi interessati per la generazione di fotoni entangled.
Per ogni dispositivo utilizzato (due laser e lo spettrometro) è stata creata una classe che ne definisce l'oggetto e uno script che si occupa di creare gli oggetti interessati e usarli per il testing degli strumenti.
Per l'Ondax's LM Series Compact Laser Module e lo spettrometro sono state create, inoltre, due classi che fanno uso dell'interfaccia grafica rispettivamente per configurare in un particolar modo il laser e visualizzare lo spettro di questo tramite lo spettrometro.

4.1 Sviluppo

Lo sviluppo delle classi e degli script è stato fatto in linguaggio Python [17], che al giorno d'oggi sta riscuotendo particolare successo per le sue caratteristiche di alto livello nonostante prestazioni computazionali inferiorri rispetto ad altri linguaggi come C. Python però permette un'estrema flessibilità e comprensione, tramite l'uso di moltissime librerie capaci di interagire con altrettanti dispositivi con l'uso di oggetti, senza doversi preoccupare di tipizzazione delle variabili, separazione di blocchi di codice tramite parentesi (viene utilizzata l'indentazione) e memoria dinamica. La sua semplicità ha permesso un'associazione ad un comune pseudo-codice, ed effettivamente è possibile prendere confidenza facilmente con la sintassi e la lettura del codice.
La struttura delle classi che definiscono gli oggetti si articola in modo tale da favorire l'ereditarietà tra gli oggetti creati: è presente una classe astratta generica `Device`, che rappresenta un qualsiasi tipo di componente che si vuole utilizzare

per raccogliere dati o effettuare test. Questa classe ha poi due figli: il primo è lo spettrometro, realizzato con una classe che eredita parzialmente i metodi astratti dichiarati nella classe padre. La seconda invece è un'altra classe astratta `LaserDevice` che ha il compito di rappresentare un generico dispositivo laser in cui verranno elencati i metodi astratti comuni per entrambi i laser di pompa. Questi saranno rappresentati come ulteriori figli della classe `LaserDevice`, ed andranno a specificare eventuali metodi aggiuntivi disponibili per il singolo dispositivo.
Si veda di seguito il diagramma UML che rappresenta la struttura adottata:

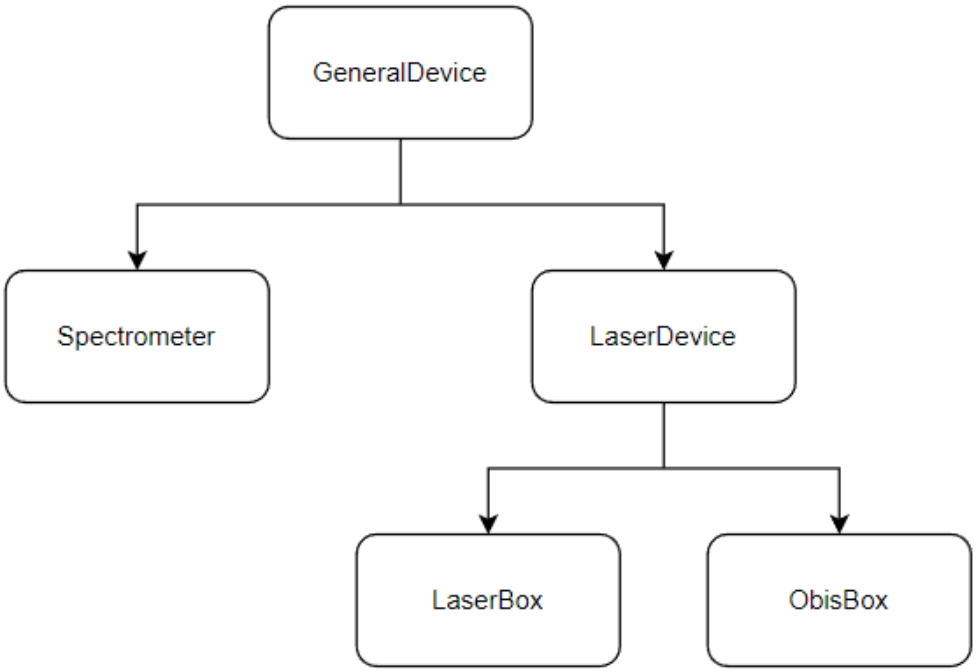

Figura 4.1: Diagramma UML della struttura

4.1.1 Laser classes

Le due classi figlie della classe `LaserDevice` sono relative alla specifica descrizione dei due laser di pompa utilizzati per la sperimentazione. Questi comunicano con il computer con un cavo seriale e un cavo USB attraverso i quali viene stabilita la connessione. Python si occupa della gestione della comunicazione attraverso l'import del modulo `pyserial` [18] per il primo e di `easy_scpi` [19] per il secondo, che permettono l'uso e la configurazione della porta seriale del computer in utilizzo.

Tramite l'utilizzo di `tkinter` [20] e `pyqtgraph` [21], altri moduli che permettono l'utilizzo di metodi e oggetti direttamente da python, è stata creata una classe che gestisce il laser tramite un'interfaccia grafica (`laserGUI`). La classe permette di accendere e spegnere il laser, impostare corrente e temperatura e farsi ritornare i valori dei parametri periodicamente con un timer. La finestra che viene generata in esecuzione è composta da campi di testo, label e bottoni che permettono di eseguire le varie azioni etichettate.

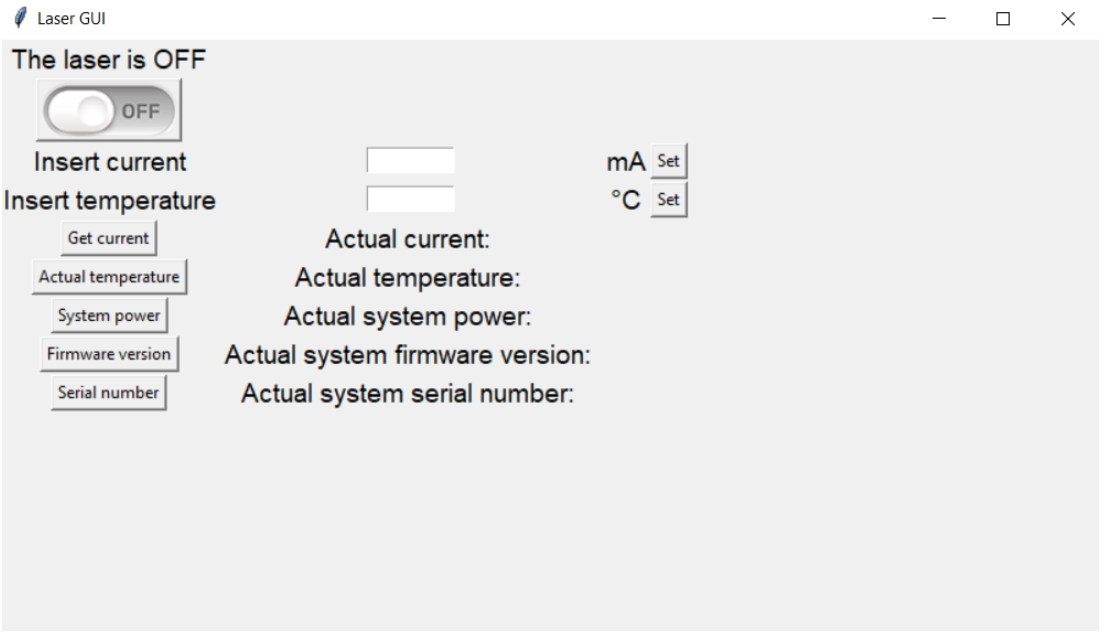

Figura 4.2: Interfaccia grafica con cui controllare il laser

4.1.2 Spectrometer class

Anche per lo spettrometro è stata scritta una classe che si occupa della struttura e una che si occupa della creazione e gestione dell'oggetto. In particolare tramite l'utilizzo del modulo Seabreeze [22] è possibile comunicare tramite comandi con lo spettrometro.
La classe `spectroGUI`, che si occupa della creazione dell'oggetto, è un'interfaccia molto semplice in cui è possibile impostare il tempo d'integrazione ed effettuare il plot del grafico

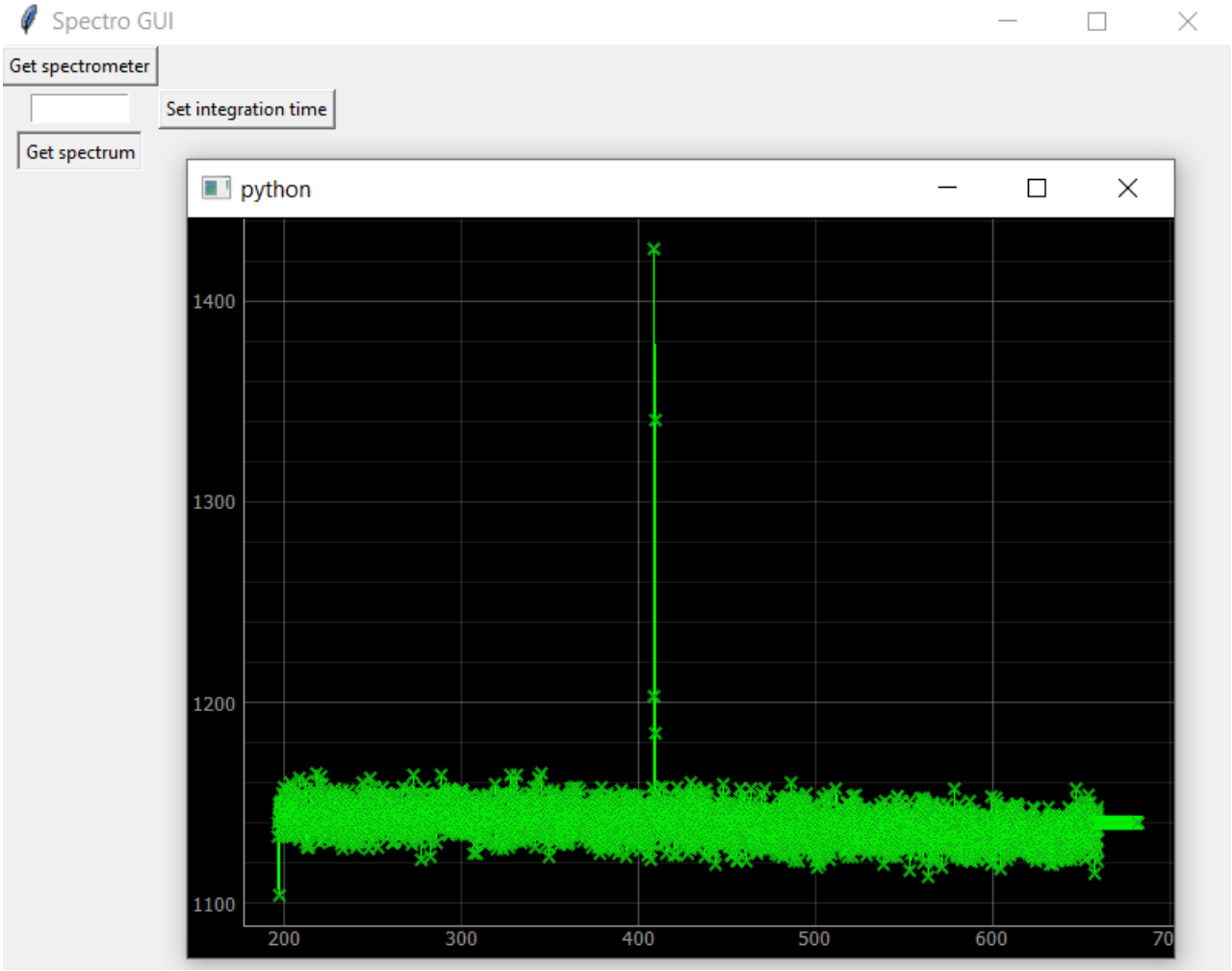

Figura 4.3: Interfaccia grafica con cui controllare lo spettrometro

A scopo più pratico è stato poi sviluppato uno script in grado di istanziare entrambi i tipi di oggetti che comandano l'architettura e di controllarli tramite i comandi in seriale illustrati nell'appendice. La classe si occupa anche della raccolta dei dati.
Per approfondire la struttura delle classi si veda l'appendice A

4.2 Raccolta dati

Al fine di ottenere i dati relativi alle prestazioni dei componenti in uso, è stato creato uno script che permette di settare i parametri del laser di pompa per poi controllare e ottenere i valori emessi con lo spettrometro.
In particolare, dopo aver istanziato il laser e lo spettro, vengono controllati i valori iniziali di corrente e temperatura; successivamente alcune variabili delimitano

il range per cui i componenti operano. Tramite due cicli for annidati si scorrono i range precedentemente impostati, facendo scalare la temperatura di 0.1 °C (da 20 a 30 gradi) per ogni valore di corrente (da 40 a 60 mA) o potenza (da 5 a 38 mW). Dopo aver impostato correttamente i valori, si attende tramite l'uso un ciclo while che il laser raggiunga la temperatura e la corrente (o la potenza) indicata, ci vuole infatti, qualche manciata di secondi affinché vengano soddisfatti i range di tolleranza presenti nel ciclo. Viene simultaneamente controllato se lo spettrometro è stato saturato nel corso del cambiamento dei parametri, nel caso, il laser viene immediatamente spento per evitare di danneggiare il dispositivo.
A questo punto vengono dichiarate delle variabili, alle quali viene assegnato subito il valore della lunghezza d'onda e delle intensità rilevate dallo spettrometro e che li rappresentano tramite una lista. Queste liste vengono ridotte per ottimizzare i dati e salvare solamente quelli che riguardano una lunghezza d'onda tra i 390 e i 420 nm.
Successivamente salviamo anche i valori attuali dei laser, salviamo poi in un archivio `.npz`, tramite l'utilizzo di `numpy (numpy.savez)` i dati relativi alle variabili appena istanziate. Gli archivi sono leggibili tramite un altro comando della libreria `numpy (numpy.load)` e possono essere usati per effettuare varie operazioni.
Al termine dello script è possibile chiamare altri metodi per verificare il corretto funzionamento del software e dei vari componenti, come il plot. Da precisare che lo script è relativo ai metodi di entrambi i laser è dunque necessario creare due script diversi per la singola gestione della raccolta dei dati.

4.3 Analisi e confronto

Una volta ottenuti i file con all'interno le strutture contenenti i valori delle intensità e delle lunghezze d'onda cercate, è possibile eseguire ulteriori operazioni.
Con lo sviluppo di ulteriori script in Python è stato possibile eseguire dei plot che mostrano il comportamento dei laser in seguito dei parametri (temperatura e corrente o potenza) impostati. Dopo aver raccolto i dati di intensità, lunghezza d'onda e relativi indici negli array, vengono calcolati, tramite una funzione che esegue il fit gaussiano, i valori di altezza, media e larghezza del picco; valori che sono basati su una media del rumore ottenuto al di fuori delle lunghezze d'onda operative dei laser. Con l'utilizzo della libreria `matplotlib` è stato possibile rappresentare graficamente, per una singola misura, il grafico approssimato con i

valori ottenuti dagli array e quello con il fit gaussiano a seguito della chiamata a funzione `curve_fit` della libreria `scipy.optimize`.

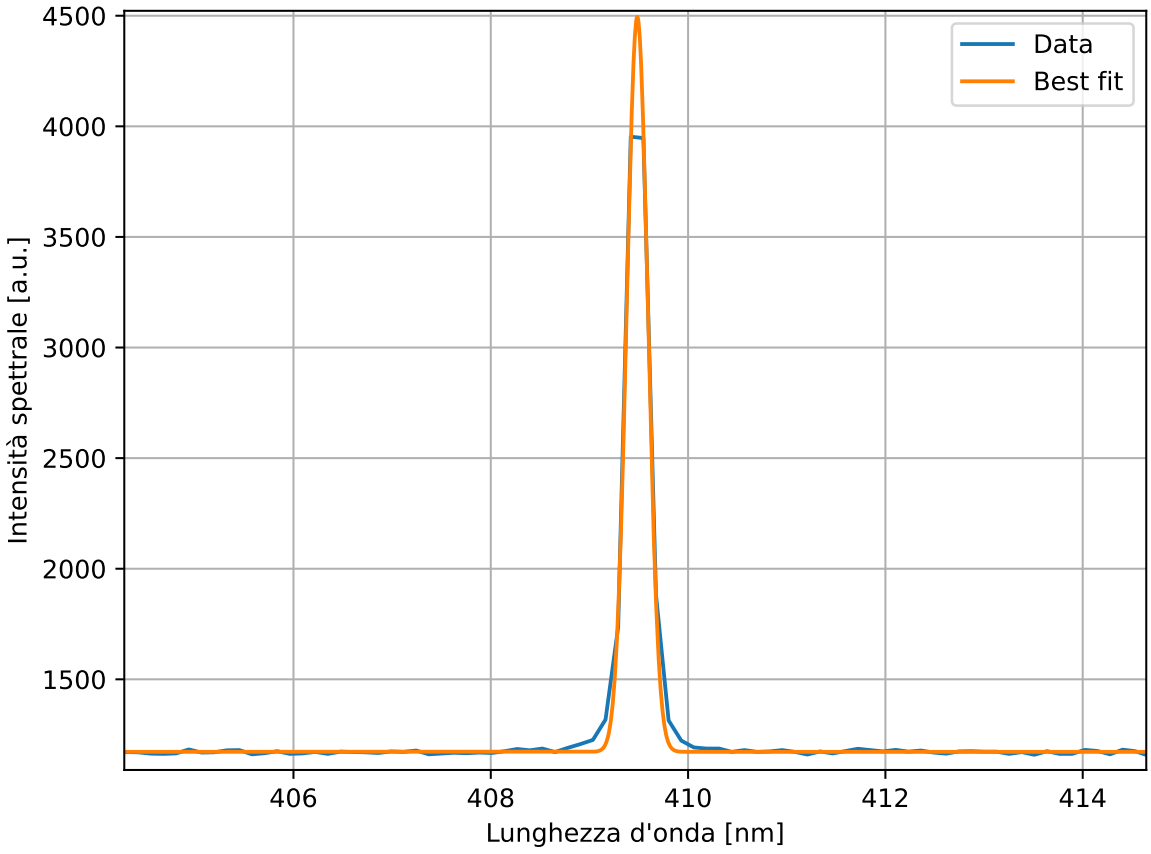

Figura 4.4: Plot di una singola misura, si noti la differenza tra le due curve

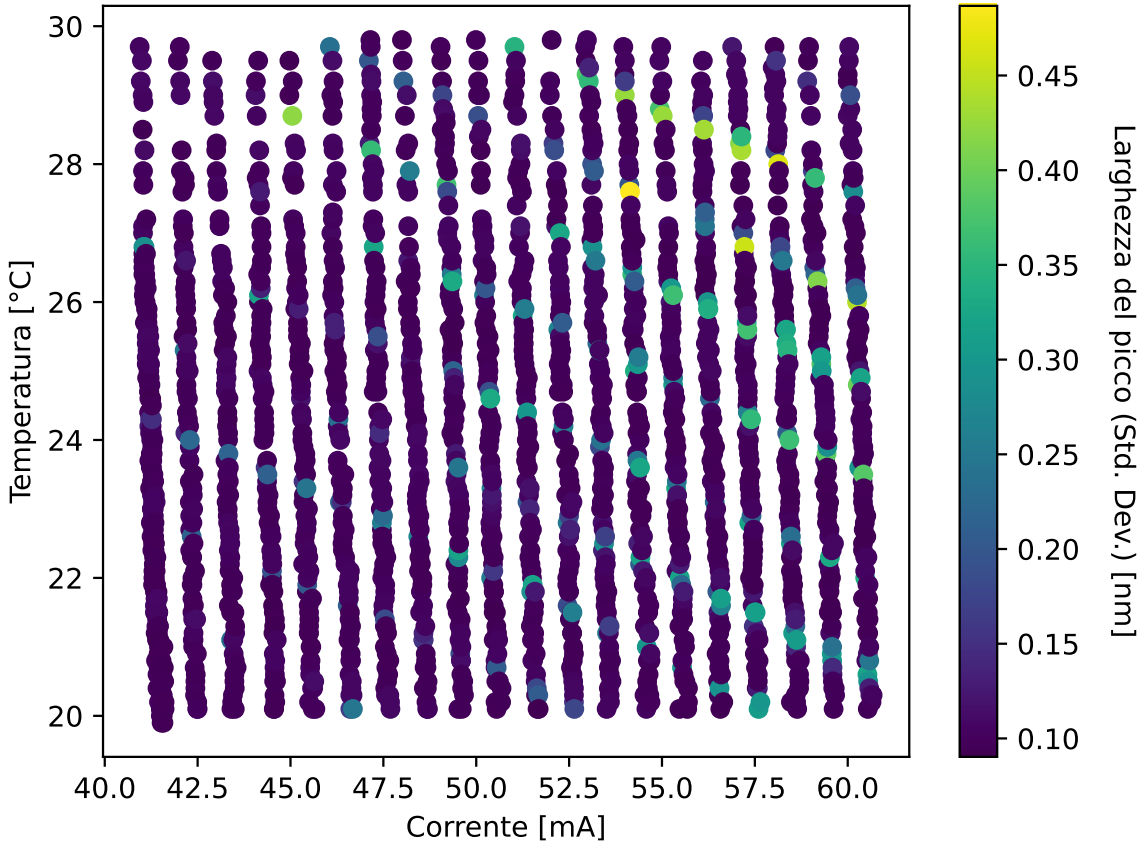

Figura 4.5: Larghezza del picco del primo laser

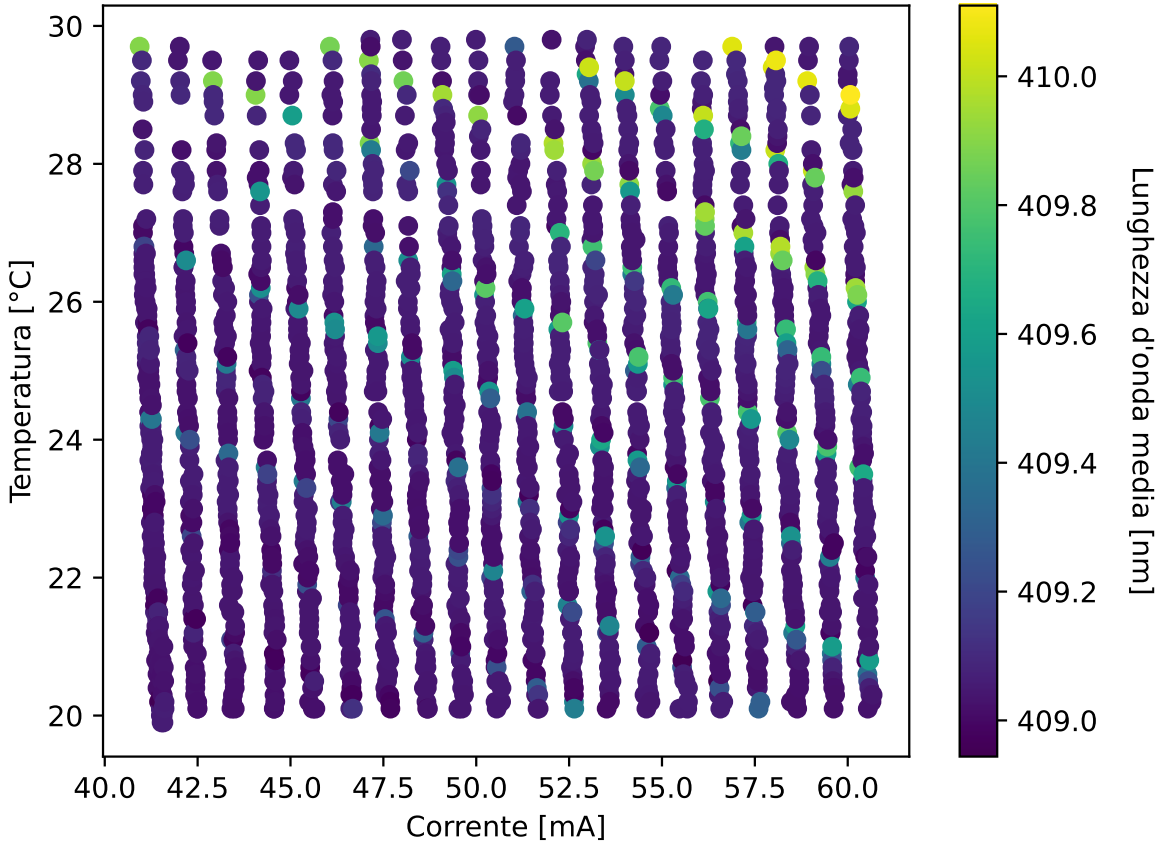

Figura 4.6: Lunghezza d'onda media del primo laser

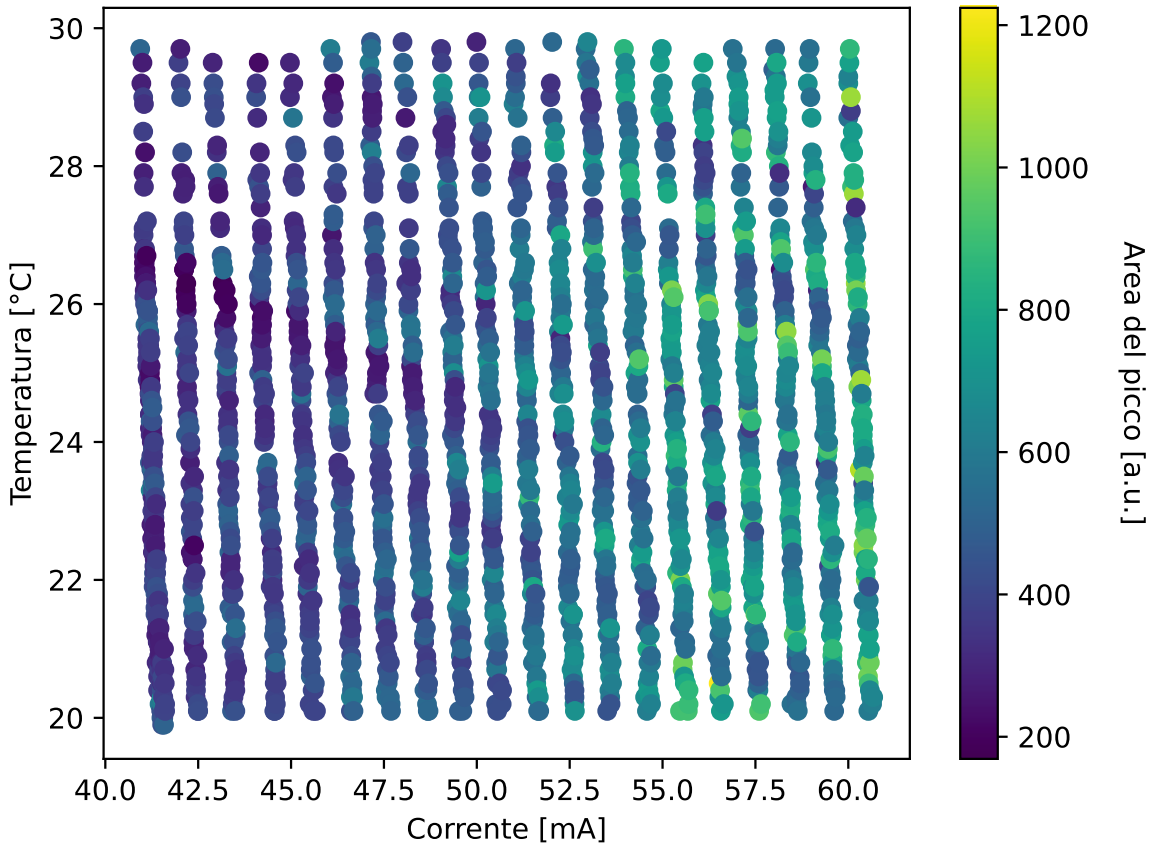

Figura 4.7: Area del picco del primo laser

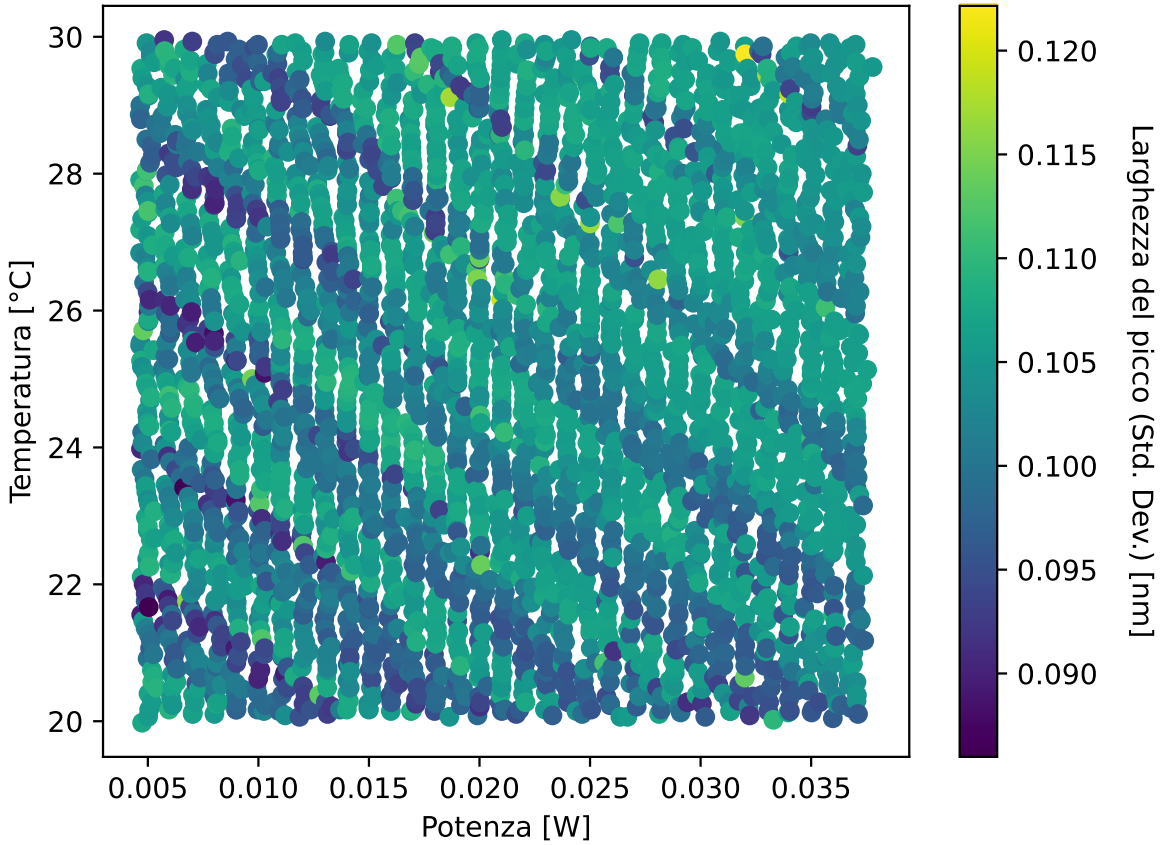

Figura 4.8: Larghezza del picco del secondo laser

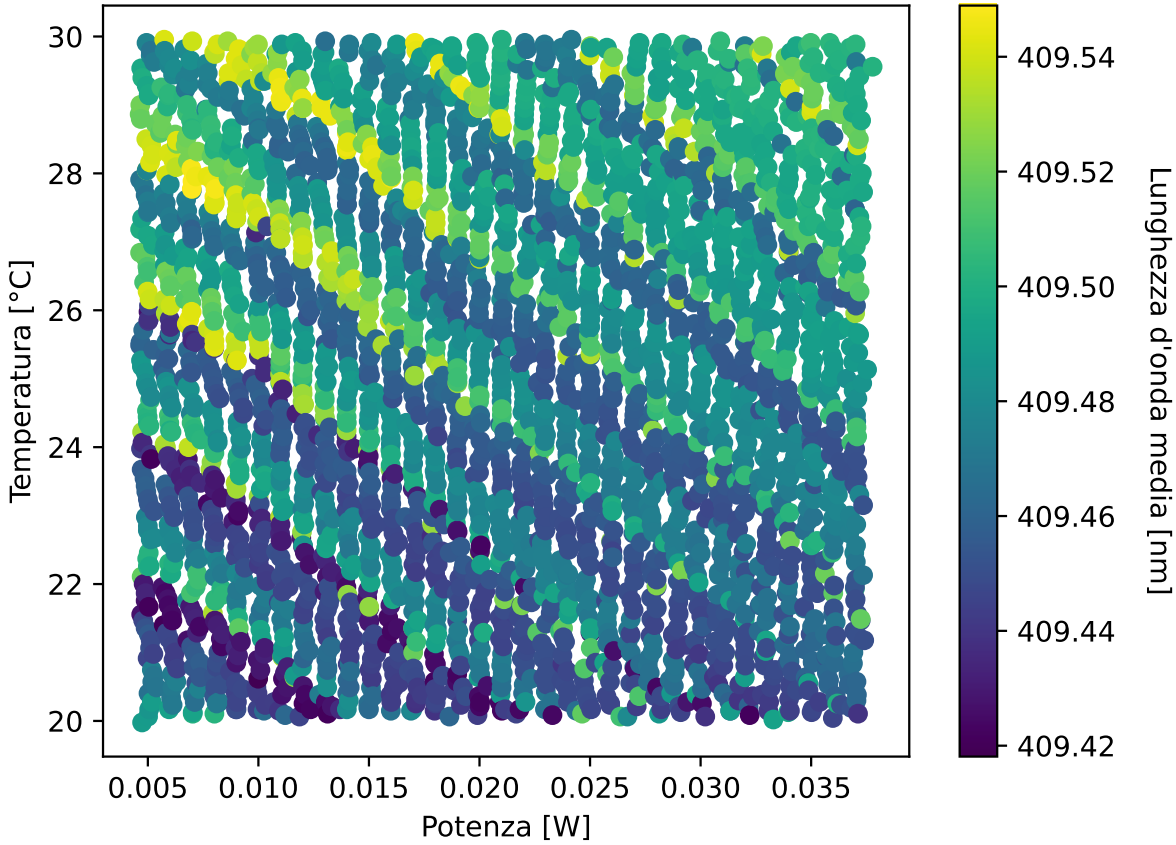

Figura 4.9: Lunghezza d'onda media del secondo laser

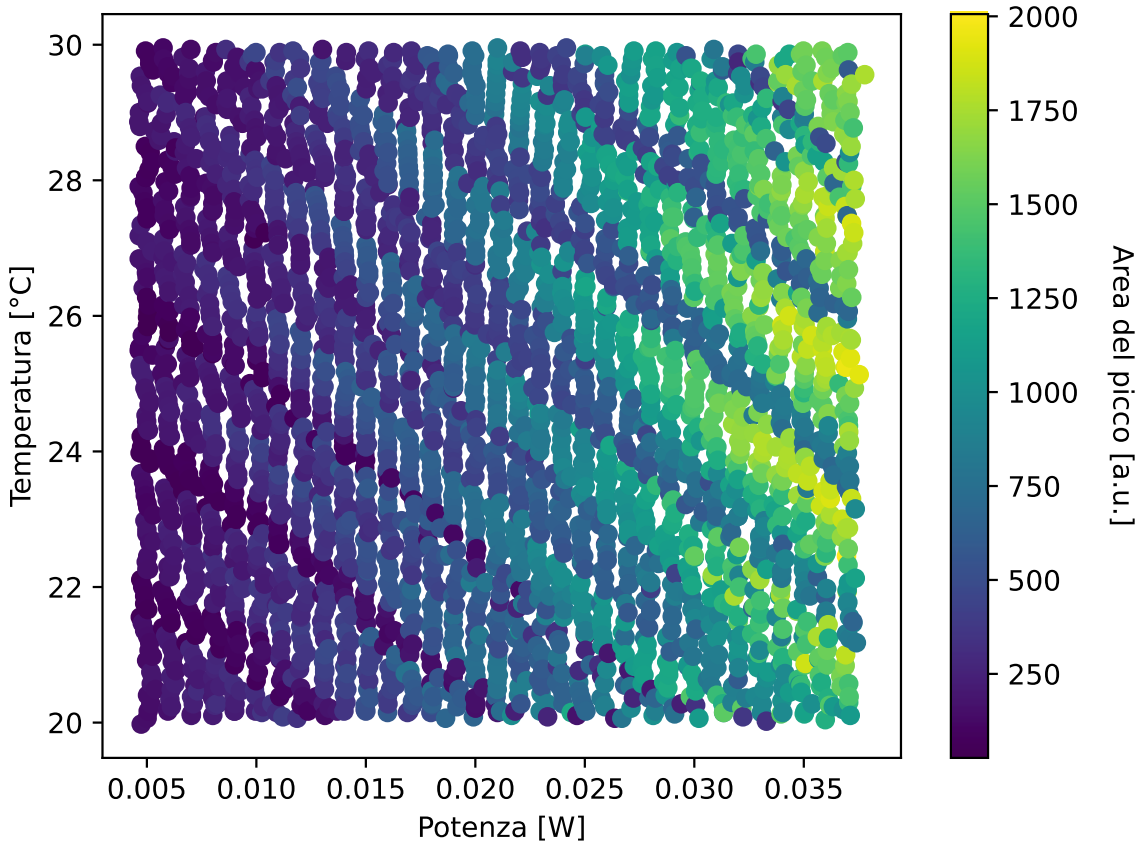

Figura 4.10: Area del picco del secondo laser

Osservando i grafici si nota come per entrambi laser siano presenti delle fasce di colore uniforme. Esse indicano che le proprietà variano più lentamente se a seguito di un aumento di potenza (o corrente) si fa eseguire una diminuzione della temperatura. Viceversa, muovendosi nell'altra direzione (ad esempio aumentando sia potenza che temperatura) le proprietà cambiano più velocemente e mostrano oscillazioni. Si vede però che sul laser Ondax la scala dei valori è decisamente maggiore (per i primi due grafici) e questo ne indica una maggiore varianza rispetto all'OBIS: : il laser ondax mostra alcune forti discontinuità nell'andamento delle proprietà per piccoli cambiamenti di temperatura o corrente (si notino ad esempio le linee di colore chiaro in Figure 4.4 e 4.5). Si è osservato che esse corrispondono a situazioni in cui lo spettro diventa bimodale, mostrando un secondo picco di poco spostato rispetto a quello principale. Corrispondentemente, la frequenza di produzione e raccolta di coppie cala di molto: a titolo di esempio si è osservato il punto a 54.1mA e 27.8°C (quello che mostra la larghezza maggiore in Figura 4.4) e si sono misurate quattro volte meno coppie rispetto alla configurazione con soli 0.2°C in meno. Infine, sempre nei grafici relativi al

laser ondax, si nota che le colonne di punti sono leggermente inclinate. Questo significa che per la stessa impostazione di corrente, il laser si stabilizza a correnti inferiori quando la temperatura è più alta. I dati mancanti nei grafici relativi al laser ondax (si veda la regione in alto a sinistra) indicano che il dispositivo non è riuscito a impostare la temperatura e corrente selezionate entro un sufficiente grado di precisione e tempo massimo. Ciò indcica una minore accuratezza dei meccanismi di regolazione interna dell'ondax rispetto all'obis, che non mostra dati mancanti.
Nel corso della sperimentazione ogni laser impiegava circa 3-4 ore per la presa dati, e per necessità di tempo è stato svolto solo un ciclo per laser, il che comunque è stato sufficiente per raccogliere un numero soddisfacente di dati.
Questo è solo un tipo di misura e confronto che è possibile effettuare; tramite python è possibile infatti fare ulteriori analisi che possono approfondire i risultati ottenuti, aggiungendo ulteriori dettagli e analizzando nel profondo alcune specifiche della sorgente.

Capitolo 5

Conclusioni

Al termine del percorso di studio e di applicazioni in laboratorio, è stato possibile sviluppare il software necessario per analizzare e caratterizzare, tramite l'utilizzo dello spettrometro, i due laser di pompa della sorgente di fotoni entangled presa in considerazione in questa tesi. Tramite l'analisi dei dati è stato possibile identificare le potenzialità dei dispositivi, verificandone le prestazioni in base ai parametri impostati durante la raccolta dati. In particolare, dopo aver collezionato due vasti dataset relativi ai due laser è stato possibile, tramite l'acquisizione di nuove conoscenze di Python, utilizzare i dati ottenuti per visualizzare graficamente i risultati, con la possibilità di approfondire lo studio per alcuni valori risultati cruciali. Lo studio di base della meccanica quantistica e della quantum information è stato utile per scoprire ed arricchire conoscenze inerenti anche in vista del prossimo percorso universitario personale che riguarderà l'approfondimento della sicurezza informatica, i cui corsi mirano anche allo studio della crittografia quantistica: una delle possibili applicazioni della generazione di fotoni entangled.

Appendice A

Struttura delle classi

Questo appendice ha l'utilità di esplicare il codice descritto nel Capitolo 4 di questa tesi. La sua funzione è di essere una API di riferimento e documentazione.

- class GeneralDevice(ABC): ha il compito di rappresentare un generico oggetto presente nella struttura, i suoi metodi vengono ereditati da tutti i device presenti.

```
def __init__(self) -> None:
    pass
'''
@abstractmethod
Costruttore della classe astratta
'''


def powerOff(self) -> None:
    pass
'''
@abstractmethod
Metodo astratto che viene ereditato da tutti i
device sottostanti alla classe padre, questi avranno
un metodo ereditato che gli permette di eseguire
uno shutdown del dispositivo
'''
```

- class Spectrometer(GeneralDevice): la classe ha il compito di descrivere accuratamente le funzioni dello spettrometro Ocean Optics, questo avviene

tramite l'import del modulo seabreeze. Viene importato anche il modulo tkinter e poter creare un'interfaccia grafica per la gestione manuale del laser. La classe eredita i due metodi della classe padre, definendoli.

```
def __init__(self) -> None:
    self.spectrometer = Spectro.from_first_available()
    '''
    Il costruttore crea un oggetto Spectrometer dal primo
    spettrometro rilevato connesso
    e disponibile per l'utilizzo

    @param self: l'oggetto Spectrometer
    '''

def powerOff(self) -> None:
    self.spectrometer.close()
    '''
    Chiude la connessione al dispositivo Seabreeze

    @param self: l'oggetto Spectrometer
    '''

def printDevices(self) -> None:
    """Print devices"""
    print(list_devices())
    '''
    Stampa la lista dei dispositivi connessi
    supportati da Ocean Optics

    @param self: l'oggetto Spectrometer
    '''

def getSpectrometer(self) -> Spectrometer:
    return self.spectrometer
    '''
    Restituisce l'oggetto Spectrometer
```

```
        @param self: l'oggetto Spectrometer
        '''

    def setIntegrationTime(self, time, label : Label) -> None:
        '''
        Il metodo imposta un tempo di integrazione
        dello spettrometro secondo il quale il dispositivo
        misura il fascio luminoso.

        @param self: l'oggetto Spectrometer
        @param time: tempo di integrazione in microsecondi
        @param label: l'oggetto Label che visualizza
        il tempo di integrazione
        '''

    def setIntegrationTimeScript(self,time):
        '''
        Il metodo imposta un tempo di integrazione
        dello spettrometro secondo il quale il dispositivo
        misura il fascio luminoso.

        @param self: l'oggetto Spectrometer
        @param time: tempo di integrazione in microsecondi

        @return: tempo di integrazione in microsecondi
        '''

    def getWaveLength(self) -> NDArray[numpy.float_]:
        '''
        Il metodo restituisce la lunghezza d'onda in nanometri

        @param self: l'oggetto Spectrometer

        @return: la lunghezza dell'onda in nanometri
        '''
```

```
def getIntensities(self) -> NDArray[numpy.float_]:
    '''
    Il metodo restituisce l'intensità del fascio luminoso
    misurato dallo spettrometro

    @param self: l'oggetto Spectrometer

    @return: l'intensità del fascio luminoso misurato
    dallo spettrometro
    '''

def getSpectrum(self) -> NDArray[numpy.float_]:
    '''
    Il metodo restituisce l'intensità del fascio luminoso
    misurato dallo spettrometro
    e la lunghezza d'onda in nanometri

    @param self: l'oggetto Spectrometer

    @return: l'intensità del fascio luminoso misurato
    dallo spettrometro e la lunghezza d'onda in nanometri
    '''

def isSaturated(self) -> Bool:
    '''
    Il metodo restituisce True se
    lo spettrometro è saturato,
    False altrimenti

    @param self: l'oggetto Spectrometer

    @return: True se
    lo spettrometro è saturato, False altrimenti
    '''

def plotSpectrum(self) -> None:
```

```
        '''
        Questo metodo permette di creare a schermo
        una nuova finestra sulla quale viene
        effettuato il plot dello spettro rilevato
        dello spettrometro.
        Sugli assi x e y vengono rappresentati
        i rispettivi valori di
        lunghezze d'onda e intensità,
        questi vengono aggiornati
        tramite la chiamata periodica
        della funzione upgrade(self,params),
        in cui i parametri passati sono il
        grafico e la linea che
        rappresenta le intensità dello spettro.

        @param self: l'oggetto Spectrometer
        '''

    def upgrade(self, params) -> None:
        '''
        Il metodo permette di aggiornare il grafico generato
        dalla funzione plotSpectrum
        (che la richiama ogni secondo) tramite
        la sovrascrittura dei dati impostati nel disegno.

        @param self: l'oggetto Spectrometer
        '''
```

- class LaserDevice(GeneralDevice, ABC): la classe astratta permette di descrivere un generico dispositivo utilizzato come laser di pompa per la sperimentazione. Viene effettuato l'import del modulo serial per gestire la creazione di un oggetto figlio. La classe eredita i metodi astratti della classe padre GeneralDevice

```
    def __init__(self) -> None:
```

```
        '''
        Il costruttore crea un oggetto LaserDevice

        @param self: l'oggetto LaserDevice
        '''

    @abstractmethod
    def powerOff(self):
        pass
        '''
        Il metodo spegne il laser

        @param self: l'oggetto LaserDevice
        '''

    @abstractmethod
    def powerOn(self):
        pass
        '''
        Il metodo accende il laser

        @param self: l'oggetto LaserDevice
        '''

    @abstractmethod
    def setTemp(self,temp):
        pass
        '''
        Il metodo imposta la temperatura del laser

        @param self: l'oggetto LaserDevice
        @param temp: temperatura in gradi centigradi
        '''

    @abstractmethod
    def getCurrent(self):
```

```
        pass
        '''
        Il metodo restituisce la corrente del laser

        @param self: l'oggetto LaserDevice
        '''

    @abstractmethod
    def getTemp(self):
        pass
        '''
        Il metodo restituisce la temperatura del laser

        @param self: l'oggetto LaserDevice
        '''

    @abstractmethod
    def getFloatTemp(self):
        pass
        '''
        Il metodo restituisce la temperatura del laser
        in gradi centigradi

        @param self: l'oggetto LaserDevice
        '''

    def readLine(self) -> None:
        '''
        Il metodo restituisce la stringa letta dal laser

        @param self: l'oggetto LaserDevice
        '''
```

- class LaserBox(LaserDevice): rappresenta la classe che gestisce il laser Ondax's LM Series Compact Laser Module, eredita i metodi della classe

LaserDevice e ne consente l'utilizzo

```
def __init__(self) -> None:
    '''
    Il costruttore crea un oggetto LaserBox dal primo
    laser rilevato connesso e disponibile per l'utilizzo

    @param self: l'oggetto LaserBox
    '''

# Overriding abstractmethod
def powerOn(self) -> None:
    '''
    Il metodo accende il laser

    @param self: l'oggetto LaserBox
    '''

# Overriding abstractmethod
def powerOff(self) -> None:
    '''
    Il metodo spegne il laser

    @param self: l'oggetto LaserBox
    '''

# Overriding abstractmethod
def setCurrent(self,current,minCurrent,maxCurrent) -> String:
    '''
    Il metodo imposta la corrente del laser

    @param self: l'oggetto LaserBox
    @param current: corrente in mA
    @param minCurrent: corrente minima
    @param maxCurrent: corrente massima

    @return: stringa di conferma
```

```
    '''

# Overriding abstractmethod
def setTemp(self,temp,minTemp,maxTemp) -> String:
    '''
    Il metodo imposta la temperatura del laser

    @param self: l'oggetto LaserBox
    @param temp: temperatura in gradi centigradi
    @param minTemp: temperatura minima
    @param maxTemp: temperatura massima

    @return: stringa di conferma
    '''

# Overriding abstractmethod
def getCurrent(self) -> String:
    ''''
    Il metodo restituisce la corrente del laser

    @param self: l'oggetto LaserBox

    @return: corrente in mA
    '''

# Overriding abstractmethod
def getTemp(self) -> String:
    '''
    Il metodo restituisce la temperatura del laser

    @param self: l'oggetto LaserBox

    @return: temperatura in gradi centigradi
    '''

# Overriding abstractmethod
```

```
def getFloatCurrent(self) -> Float:
    '''
    Il metodo restituisce la corrente del laser

    @param self: l'oggetto LaserBox

    @return: corrente in mA
    '''

def getFloatTemp(self) -> Float:
    '''
    Il metodo restituisce la temperatura del laser

    @param self: l'oggetto LaserBox

    @return: temperatura in gradi centigradi
    '''

def readLine(self) -> String:
    '''
    Il metodo legge una riga dal laser

    @param self: l'oggetto LaserBox

    @return: stringa letta
    '''
```

- class OBISBox(LaserDevice): rappresenta la classe che gestisce il laser Coherent Obis LX, eredita i metodi della classe LaserDevice e ne consente l'utilizzo. Questo dispositivo non utilizza una connessione seriale, viene importato infatti un modulo diverso: easy_scpi.

```
def __init__(self) -> None:
    '''
    Il costruttore crea un oggetto LaserBox dal primo
```

```
        laser rilevato connesso e disponibile per l'utilizzo

        @param self: l'oggetto LaserBox
        '''

    # Overriding abstractmethod
    def printDevices(self) -> Any:
        '''
        Il metodo stampa i dispositivi connessi

        @param self: l'oggetto LaserBox

        @return: stringa di conferma
        '''

    def checkHandshake(self) -> String:
        '''
        Il metodo controlla lo stato dell'handshake

        @param self: l'oggetto LaserBox

        @return: stringa di conferma
        '''

    # Overriding abstractmethod
    def powerOn(self) -> None:
        '''
        Il metodo accende il laser

        @param self: l'oggetto LaserBox

        @return: stringa di conferma
        '''

    # Overriding abstractmethod
    def powerOff(self) -> None:
```

```
        '''
        Il metodo spegne il laser

        @param self: l'oggetto LaserBox

        @return: stringa di conferma
        '''

    def getWavelength(self) -> Any:
        '''
        Il metodo restituisce la lunghezza d'onda del laser

        @param self: l'oggetto LaserBox

        @return: stringa di conferma
        '''

    # Overriding abstractmethod
    def setTemp(self,temp,minTemp,maxTemp) -> None:
        '''
        Il metodo imposta la temperatura del laser

        @param self: l'oggetto OBISBox
        @param temp: temperatura da impostare
        @param minTemp: temperatura minima
        @param maxTemp: temperatura massima

        @return: stringa di conferma
        '''

    def setPower(self,power,minPower,maxPower) -> None:
        '''
        Il metodo imposta la potenza del laser

        @param self: l'oggetto LaserBox
        @param power: potenza da impostare
```

```
        @param minPower: potenza minima
        @param maxPower: potenza massima

        @return: stringa di conferma
        '''

    def getPower(self) -> Int:
        '''
        Il metodo restituisce la potenza del laser

        @param self: l'oggetto OBISBox

        @return: stringa di conferma
        '''

    def getFloatPower(self) -> Float:
        '''
        Il metodo restituisce la potenza del laser in float

        @param self: l'oggetto OBISBox

        @return: potenza del laser in floating point
        '''

    # Overriding abstractmethod
        '''
        Il metodo restituisce la corrente del laser

        @param self: l'oggetto OBISBox

        @return: corrente del laser
        '''

    # Overriding abstractmethod
    def getFloatCurrent(self) -> Float:
        '''
```

```
        Il metodo restituisce la corrente del laser in float

        @param self: l'oggetto OBISBox

        @return: corrente del laser in floating point
        '''

    # Overriding abstractmethod
    def getTemp(self) -> Int:
        '''
        Il metodo restituisce la temperatura del laser

        @param self: l'oggetto OBISBox

        @return: temperatura del laser
        '''

    # Overriding abstractmethod
    def getFloatTemp(self) -> Float:
        '''
        Il metodo restituisce la temperatura del laser in float

        @param self: l'oggetto OBISBox

        @return: temperatura del laser in floating point
        '''

    def getState(self) -> String:
        '''
        Il metodo restituisce lo stato del laser

        @param self: l'oggetto OBISBox

        @return: stringa di conferma dello stato del laser
        '''
```

- OBISSpectroScript e laserSpectroScript: script che utilizzano moduli e librerie creati in precedenza per l'istanziazione di oggetti e il loro uso. Dopo la creazione degli oggetti in entrambe le classi, queste si occupano di impostare correttamente i valori dei parametri dei laser, aspettando fino a quando questi non siano rientrati in un range di tolleranza. Durante questo periodo di attesa viene costantemente controllato il livello di saturazione dello spettrometro: nel caso in cui venisse raggiunto un livello superiore al 90% il laser viene spento con il metodo powerOff() (o alternativamente attenuato). Successivamente vengono salvate due liste contenenti le lunghezze d'onda e le intensità rilevate dallo spettrometro, queste vengono "tagliate" intorno a 405 nm (in cui operano entrambi i laser). I dati vengono poi salvati in file archivio .npz nominati con il valore della corrente (o potenza) e temperatura della misura. Alla fine dello script lo spettrometro viene disconnesso e il laser viene spento tramite i metodi sviluppati.

Bibliografia

[1] Processing of Mesoscopic Time-pulsed Entangled Optical fields, https://cordis.europa.eu/article/id/89563-entangling-photons-to-unravel-quantum-computing-mysteries/it.

[2] Quantum Physics - Michel le Bellac https://www.cambridge.org/core/books/quantum-physics/9A3A0754B265D451C931E0C98E6C1ED9

[3] Momento angolare dei fotoni, https://www.lescienze.it/news/2012/11/07/news/entanglement_fotoni_momento_angolare_orbitale-1353576/.

[4] Quantum numbers, https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Quantum_Mechanics/10%3A_Multi-electron_Atoms/Quantum_Numbers_for_Atoms

[5] Spazio di Hilbert https://en.wikipedia.org/wiki/Hilbert_space

[6] Entanglement https://it.wikipedia.org/wiki/Entanglement_quantistico

[7] Quantum Entanglement, https://www.youtube.com/watch?v=fkAAbXPEAtU

[8] Random number generation, https://journals.aps.org/pra/abstract/10.1103/PhysRevA.103.062206

[9] SPDC, https://www.youtube.com/watch?v=xSAoWrD03FQ

[10] SPDC, https://en.wikipedia.org/wiki/Spontaneous_parametric_down-conversion

[11] Waveplate, https://en.wikipedia.org/wiki/Waveplate

[12] T. Kim, M. Fiorentino, and F. N. C. Wong, “Phase-stable source of polarizationentangled photons using a polarization Sagnac interferometer”, https://journals.aps.org/pra/abstract/10.1103/PhysRevA.73.012316

[13] M. Schiavon ”Space Quantum Communication” https://www.research.unipd.it/handle/11577/3422779?1/tesiSchiavonReviewed.pdf

[14] Laser di pompa, https://www.coherent.com/resources/datasheet/lasers/LM_Series_404-180mW.pdf

[15] Laser di pompa, https://www.coherent.com/lasers/cw-solid-state/obis-ls-lx

[16] Spettrometro, https://www.oceaninsight.com/products/measurement--technique-bundles/plasma-bundles/

[17] Python, https://it.wikipedia.org/wiki/Python

[18] Python pyserial module, https://pypi.org/project/pyserial/

[19] easy_scpi, https://pypi.org/project/easy-scpi/

[20] Python tkinter module, https://docs.python.org/3/library/tkinter.html

[21] Python pyqtgraph module, https://www.pyqtgraph.org/

[22] Python seabreeze module, https://python-seabreeze.readthedocs.io/en/latest/