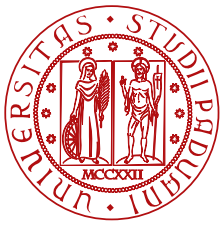


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

Ottimizzazione della generazione di stati per generatori quantistici di numeri casuali

Relatore
Dr. Avesani Marco

Laureanda
Bovo Alisea

Correlatore
Bertapelle Tommaso
Sabatini Mattia

ANNO ACCADEMICO 2023-2024

Data di laurea 27/09/2024

”If you only do what you can do,
you will never be more than who you are,
then you have to understand and believe in yourself”

Sommario

La generazione di numeri casuali riveste un ruolo cruciale in molteplici settori, come la crittografia, le simulazioni e il gioco d'azzardo. La principale sfida in questo ambito è produrre numeri che siano effettivamente casuali, evitando soluzioni che si limitino a sembrare tali. I generatori di numeri pseudocasuali, basati su algoritmi deterministici, possono infatti generare sequenze prevedibili, compromettendo la sicurezza e l'efficacia in contesti critici. Per ottenere una vera casualità, è necessario sfruttare i principi della meccanica quantistica, che fornisce un meccanismo intrinsecamente imprevedibile per la generazione di numeri casuali. In questa tesi viene presentato uno studio sull'ottimizzazione della generazione di stati per un generatore quantistico di numeri casuali o Quantum Random Number Generator (QRNG) in inglese. Dopo una breve introduzione riguardante il set-up sperimentale del QRNG, verranno descritte le procedure adottate per generare le forme d'onda necessarie alla modulazione del segnale ottico, e la conseguente creazione di uno stato quantistico. In seguito, verrà illustrato il metodo utilizzato per l'interfacciamento con gli analog to digital converters (ADCs), strumento fondamentale per la misurazione dei segnali prodotti. Infine, sarà discussa la strategia implementata per ottimizzare il segnale modulante e verranno presentati e analizzati i risultati ottenuti.

Indice

Elenco delle figure

Capitolo 1

Introduzione ai QRNG

I generatori di numeri causali (RNG, dall'inglese Random Number Generator) sono importanti per numerose applicazioni, come la crittografia, le simulazioni e il gioco d'azzardo. È possibile suddividere i RNG in due categorie: generatori di numeri pseudo-casuali (PRNG, dall'inglese Pseudo-Random Number Generator) e generatori di numeri casuali veri (TRNG, dall'inglese True Random Number Generator). I primi adoperano algoritmi deterministici per estrarre dei numeri apparentemente casuali. Partendo da una breve stringa di bit, chiamata seme (o seed in inglese), questi algoritmi la espandono e alla fine producono una stringa di bit di larghezza maggiore del seme, che segue una distribuzione uniforme. Tuttavia, i numeri generati risultano meno sicuri, poiché prevedibili: i PRNG seguono un pattern specifico e, inevitabilmente, tendono a ripetere le stesse sequenze. Per mitigare questo problema, vengono adottate strategie che riducono la ripetizione nel breve termine. Il secondo tipo, invece, misura fenomeni fisici imprevedibili, come le fluttuazioni quantistiche o il decadimento radioattivo, o di fenomeni difficili da prevedere, come i processi caotici. Un particolare tipo di TRNG è il generatore di numeri casuali quantistici (QRNG, dall'inglese Quantum Random Number Generator), che si basa sulla natura non deterministica della meccanica quantistica per garantire una vera casualità.
Il processo di generazione di un QRNG può essere suddiviso in due blocchi principali: una sorgente di entropia e una fase di post-processing. Il primo blocco consiste nella misurazione di un fenomeno fisico isolato, questo determina in parte l'affidabilità, l'entropia e le prestazioni dell'intero sistema. Dalle misure effettuate si ricava una prima stringa di bit grezza, visto che sia il processo di misurazione e di quantizzazione presentano del rumore e degli errori. Di conseguenza, la stringa ottenuta non contiene bit realmente casuali, poiché potrebbe essere parzialmente conosciuta da un avversario. Pertanto, essa viene fornita in ingresso al secondo blocco per generare una stringa di bit più corta, priva di correlazioni tra i bit e completamente scorrelata con l'avversario, garantendo così un elevato livello di sicurezza. [1]
Uno dei primi tipi di QRNG realizzati utilizzava il decadimento radioattivo come fonte di entro-

pia. I tempi di emissione delle particelle radioattive sono intrinsecamente casuali e indipendenti tra loro. Configurando adeguatamente un dispositivo di rilevamento, era possibile generare un impulso per ogni particella rilevata, consentendo così di estrarre bit casuali. Questi bit potevano essere derivati, ad esempio, dalla frequenza di arrivo delle particelle o dall'intervallo di tempo tra due particelle consecutive. [1] Tuttavia, questo approccio presentava un tasso di generazione piuttosto basso ed è stato superato a causa della difficoltà nel reperire materiali radioattivi e per le complicazioni legate al loro utilizzo. Nel tempo si sono sviluppati altri tipi di QRNG, capaci di sfruttare fenomeni quantistici più efficienti e di raggiungere tassi di generazione significativamente più elevati. Alcuni esempi includono QRNG basati su fenomeni di ottica quantistica, come la rivelazione di singoli fotoni (SPD, dall'inglese Single Photon Detection), le fluttuazioni del vuoto quantistico, l'emissione spontanea amplificata (ASE, dall'inglese Amplified Spontaneous Emission) e il rumore di fase dei laser. Questi sviluppi hanno migliorato sia l'efficienza che la praticità dei QRNG rispetto ai primi sistemi basati sul decadimento radioattivo. [1]

I QRNG possono essere classificati in tre categorie: device-independent, semi-device-independent e device-trusted QRNG.

I device-independent QRNG (DI-QRNG), come suggerisce il nome, non fanno assunzioni sui dispositivi, quindi non dipendono da alcun dispositivo specifico e solitamente si basano sulla violazione della diseguaglianza di Bell. In questa categoria il funzionamento degli strumenti e i loro componenti non sono noti a priori. Il comportamento dei dispositivi viene analizzato esclusivamente osservando le risposte generate in seguito a stimoli esterni. Tuttavia, per garantire il corretto funzionamento di questi QRNG, è necessaria un'implementazione precisa e complessa, volta a chiudere le possibili lacune nei test di Bell ("closing loopholes in Bell tests"). Questi dispositivi, a causa dell'elevata complessità, comportano costi significativi e presentano un basso tasso di generazione dei numeri casuali, limitandone l'utilizzo commerciale. [2]

I device-trusted QRNG, al contrario dei DI-QRNG, si basano sulla fiducia nei dispositivi utilizzati e richiedono una caratterizzazione completa di tutti i componenti coinvolti. Inoltre, è fondamentale avere una conoscenza approfondita della natura del fenomeno quantistico sfruttato. Questo approccio consente di raggiungere un elevato tasso di generazione di numeri casuali. Questa categoria include la maggior parte dei QRNG commerciali, che sono relativamente semplici da implementare e offrono un'elevata velocità di generazione di numeri. Tuttavia, essi risultano meno sicuri. Se si assumono completamente affidabili i dispositivi, non si prendono in considerazione le potenziali imperfezioni dei componenti, come il rumore all'interno dei laser o la banda limitata dei convertitori analogico-digitale. Questo rumore classico potrebbe essere sfruttato da un potenziale avversario per influenzare e controllare la generazione dei numeri.[3]

I semi-device-independent QRNG costituiscono un compromesso tra le due categorie precedenti, poiché permettono di ottenere alti tassi di generazione di numeri casuali, adatti a impieghi

pratici, mantenendo al tempo stesso un elevato livello di sicurezza. Rispetto ai DI-QRNG, richiedono ipotesi meno stringenti, poiché necessitano di una caratterizzazione completa solo del dispositivo di misura o della sorgente. A seconda di questa caratterizzazione, si distinguono in Source-Independent (SI) QRNG, Measurement-Device-Independent (MDI) QRNG e semi-device independent (SDI) QRNG. In questa categoria rientrano protocolli in cui non vengono caratterizzati né la sorgente né il ricevitore, ma si effettuano assunzioni su specifici aspetti del fenomeno fisico utilizzato. Tra questi, si considerano la dimensione degli stati quantistici prodotti, il grado di sovrapposizione degli stessi, la loro energia e altre proprietà simili, che consentono comunque di garantire la sicurezza e l'affidabilità del sistema senza una caratterizzazione completa dei dispositivi.[4].

Inoltre, i QRNG possono essere ulteriormente classificati in base al tipo di variabili utilizzate per l'implementazione: variabili discrete (DV) o variabili continue (CV).

Nell'ottica quantistica a variabili discrete, i dispositivi solitamente operano in regime di singolo fotone, basandosi sul principio del rilevamento di un singolo fotone. In questi sistemi, lo stato quantistico ha meno gradi di libertà, a causa della dimensione finita dello spazio di Hilbert. Inoltre, l'ottica a singolo fotone è relativamente insensibile alle perdite e richiede rivelatori di singoli fotoni, che sono dispositivi costosi.

Al contrario, gli stati ottici quantistici a variabili continue sono più sensibili alle perdite, ma offrono vantaggi quali metodi di rilevazione più economici ed efficienti. Come suggerisce il nome, in questi sistemi lo stato quantistico può assumere un continuum di valori. L'informazione è codificata nei gradi di libertà continui del campo elettromagnetico, come le quadrature, e il sistema di rivelazione richiede dispositivi omodina o eterodina, costituiti da fotodiodi classici. I QRNG a variabili continue offrono tassi di generazione superiori rispetto a quelli a variabili discrete [5] .

Nella presente tesi è stato impiegato un semi-device-independent QRNG a variabili continue, realizzato interamente in fibra ottica e utilizzando componenti Commercial Off-the-Shelf (COTS), disponibili nel mercato delle telecomunicazioni . [6]

1.1 CV-QRNG

L'obiettivo principale di questa tesi è la generazione di stati quantistici per un Quantum Random Number Generator (QRNG) basato sulla tecnica di eterodina. La generazione degli stati quantistici viene realizzata attraverso la modulazione di fase della luce emessa da un laser, che consente di produrre quattro stati distinti, analogamente a quanto avviene in una modulazione Q-PSK (Quadrature Phase Shift Keying). Successivamente, il segnale modulato viene inviato all'interno del sistema di misura eterodina, il quale permette di ottenere informazioni su entrambe le quadrature del campo elettromagnetico, rappresentate dai parametri $\hat{q}$ e $\hat{p}$. L'eterodina esegue una proiezione del campo elettrico incidente, fornendo come risultato della misurazione un valore continuo, giustificando così l'appellativo variabili continue per questo tipo di QRNG. In altri termini, il sistema è in grado di generare e misurare stati quantistici che possono assumere una continua gamma di valori, anziché essere limitati a stati discreti.
In questa sezione verranno spiegati gli elementi cardine di questa tesi presenti nel CV-QRNG.

1.1.1 Modulazione di fase

Un modulatore di fase è un dispositivo elettro-ottico che sfrutta l'effetto Pockels per modulare la fase di un'onda elettromagnetica che vi viaggia attraverso tramite una differenza di potenziale elettrico [7]. L'effetto Pockels si manifesta in diversi materiali in cui l'indice di rifrazione può essere approssimato dalla seguente relazione:

$$n(E) \approx n - \frac{1}{2} r n^3 E. \tag{1.1}$$

dove n è l'indice di rifrazione a campo elettrico nullo, r è il coefficiente elettro-ottico lineare, e E è il campo elettrico applicato. I materiali che presentano l'effetto Pockels vengono definiti celle di Pockels, e il coefficiente r solitamente assume valori compresi tra 10^{-12} - 10^{-10} m/V. Quando un fascio di luce attraversa una cella di Pockels di lunghezza L, subisce un sfasamento di fase dato da:

$$\phi = n(E) k_o L = 2\pi n(E) \frac{L}{\lambda_o},$$

dove k_o e λ_o sono il numero d'onda e la lunghezza d'onda nello spazio libero. L'applicazione di un campo elettrico E altera l'indice di rifrazione $n(E)$, modificando la fase dell'onda luminosa. Considerando l'applicazione di una tensione V tra due facce del materiale, distanti d, il campo elettrico è espresso come $E = V/d$. Utilizzando l'equazione 1.1, si ottiene lo sfasamento:

$$\phi = \phi_o - \pi \frac{r n^3 E L}{\lambda_o} = \phi_o - \pi \frac{V}{V_\pi}, \tag{1.2}$$

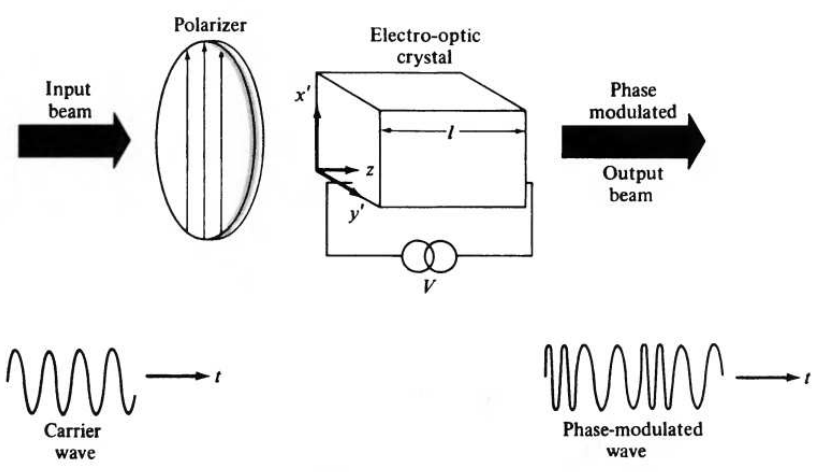


Figura 1.1: Un modulatore di fase elettro-ottico[8].

dove $\phi_o = 2\pi nL/\lambda_o$ e $V_\pi = \frac{d\lambda_o}{Lrn^3}$ è detta tensione a semi-onda (half wave voltage). V_π è la tensione necessaria per produrre uno sfasamento di π. Dall'equazione 1.2 si osserva una relazione lineare tra la tensione applicata e lo sfasamento di fase, consentendo così la modulazione della fase di un'onda ottica variando la tensione elettrica. Esistono anche modulatori elettro-ottici costruiti sotto forma di dispositivi ottici integrati, che operano a velocità più elevate e a tensioni inferiori rispetto ai dispositivi bulk. In questi sistemi, una guida d'onda è fabbricata su un substrato elettro-ottico (ad esempio $LiNbO_3$), drogato da un materiale, come il titanio (Ti), per aumentare l'indice di rifrazione. Il campo elettrico viene applicato alla guida d'onda tramite elettrodi. Dato che V_π è direttamente proporzionale alla larghezza della guida d'onda d e inversamente proporzionale alla lunghezza del materiale L e all'indice di rifrazione n, in particolari configurazioni dove $d << L$, la tensione a semi-onda può essere di pochi volt. Inoltre la luce può essere convenientemente accoppiata dentro e fuori da questi dispositivi utilizzando l'uso di fibre ottiche. [8] [7]

1.1.2 Q-PSK

La Phase shift keying (PSK) è un tipo di modulazione digitale in cui l'informazione viene codificata nella fase della portante. Nelle telecomunicazioni, la portante è un'onda elettromagnetica o un segnale elettrico, generalmente di tipo sinusoidale, caratterizzato da frequenza, ampiezza e fase note. Questo segnale viene modificato da un segnale modulante, solitamente contenente informazioni, per consentirne la trasmissione attraverso etere o cavo. La generica forma d'onda trasmessa a una frequenza f_o è descritta dalla seguente espressione generale:

$$s_n(t) = h_{Tx}(t)\cos(2\pi f_0 t + \phi_n) \quad n = 1, ..., M \tag{1.3}$$

dove $h_{Tx}(t)$ è una generica forma d'onda in trasmissione e la fase è:

$$\phi_n = \frac{\pi}{M}(2n-1).$$

In questo modo, scegliendo uno degli M valori possibili per la fase della sinusoide, si ottengono i segnali desiderati. L'informazione viene quindi codificata nei valori discreti della fase, che varia in funzione dei bit o delle sequenze di bit da trasmettere. Un'alternativa espressione alla 1.3 è data da:

$$s_n(t) = Re[h_{Tx}(t)e^{j(2\pi f_0 t+\phi_n)}] = h_{Tx}(t)(\cos(\phi_n)\cos(2\pi f_0 t) - \sin(\phi_n)\sin(2\pi f_0 t)) \quad (1.4)$$

Di conseguenza, la rappresentazione vettoriale delle forme d'onda PSK è:

$$s_n = \sqrt{E_n}[\cos(\phi_n), \sin(\phi_n)], \quad (1.5)$$

dove $E_n = ||s_n||^2 = \frac{E_h}{2}$, rappresenta l'energia del segnale. La raffigurazione vettoriale di un insieme di M segnali viene definita costellazione e nella modulazione PSK tutti i segnali trasmessi hanno la stessa energia.
Nel caso in cui $M = 4$, la modulazione prende il nome di quadrature PSK (Q-PSK) e i 4 valori della fase sono: $\phi_1 = \pi/4$, $\phi_2 = 3\pi/4$, $\phi_3 = 5\pi/4$, $\phi_4 = 7\pi/4$. La relativa costellazione si presenta:

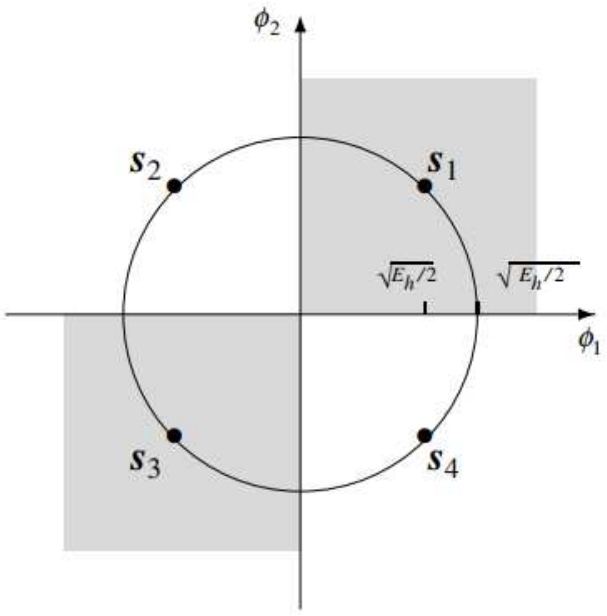


Figura 1.2: Costellazione della Q-PSK

Nella figura 1.2 le basi sono date da: $\phi_1(t) = h_{Tx}(t)\cos(2\pi f_0 t)/\sqrt{E_n}$ e $\phi_1(t) = -h_{Tx}(t)\sin(2\pi f_0 t)/\sqrt{E_n}$. [9]
In questa tesi, il metodo di modulazione Q-PSK è stato impiegato per modulare il segnale. Tuttavia, diversamente dalla modulazione digitale tradizionale, ai simboli della costellazione non sono associati bit, ma stati quantistici, utilizzati per la generazione di numeri casuali quantistici.

1.1.3 Misura eterodina

Prima di introdurre la misura eterodina, è necessario delineare alcuni principi fondamentali. Il libro [10] è stato impiegato come riferimento per questo paragrafo e per una trattazione più dettagliata, si rimanda al testo stesso. Nel presente contesto, per semplicità, si assume che la costante di Planck sia $\hbar = 1$ e si postula che il campo elettrico della luce sia dato da:

$$\hat{E} = u^*(x,t)\hat{a} + u(x,t)\hat{a}^\dagger \tag{1.6}$$

dove $u(x,t) = u_o e^{i(kx-\omega t)}$ è una semplice onda piana, mentre $\hat{a}$ è un operatore di annichilazione bosonico e obbedisce alla seguente relazione:

$$[\hat{a}, \hat{a}^\dagger] = 1 \tag{1.7}$$

L'operatore numero di fotoni (photon-number) indica il numero di fotoni presenti in un preciso modo spazio-temporale ed è definito:

$$\hat{n} \equiv \hat{a}^\dagger \hat{a}. \tag{1.8}$$

Gli operatori quadrature, indicati con le lettere $\hat{q}$ e $\hat{p}$, sono analoghi rispettivamente come la posizione e il momento dell'oscillatore elettromagnetico. Secondo il principio di indeterminazione di Heisenberg [11], indicato nella seguente formula:

$$\Delta p \Delta q \geq \frac{\hbar}{2} \tag{1.9}$$

questi operatori non commutano, ciò comporta che non è possibile misurarli simultaneamente con precisione arbitraria, infatti $[\hat{q}, \hat{p}] = i$. Le quadrature corrispondono alla parte reale e immaginaria dell'ampiezza complessa di $\hat{a}$ moltiplicata per una costante:

$$\hat{q} = 2^{-1/2}(\hat{a}^\dagger + \hat{a}) \quad \hat{p} = i2^{-1/2}(\hat{a}^\dagger - \hat{a}) \tag{1.10}$$

Inoltre l'operatore di annichilazione bosonico si può riscrivere $\hat{a} = 2^{-1/2}(\hat{q} + i\hat{p})$.
Si definiscono gli autostati $|q\rangle$ e $|p\rangle$ delle quadrature $\hat{q}$ e $\hat{p}$ stati di quadratura, tali per cui

soddisfano:

$$\hat{q}|q\rangle = q|q\rangle \quad \hat{p}|p\rangle = p|p\rangle \tag{1.11}$$

L'autostato dell'operatore numero di fotoni si indica con $|n\rangle$ e si chiama stato di Fock e soddisfa:

$$\hat{n}|n\rangle = n|n\rangle \tag{1.12}$$

Il laser adoperato in questa tesi emette stati coerenti, i quali sono gli autostati dell'operatore di annichilazione $\hat{a}$:

$$\hat{a}|\alpha\rangle = \alpha|\alpha\rangle \tag{1.13}$$

dove α è un numero complesso e lo stato coerente può essere scritto:

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \tag{1.14}$$

La misura eterodina è anche definita misura omodina a 8 porte. Pertanto, sarà preliminarmente descritta la misura omodina, seguita dalla spiegazione della misura eterodina.

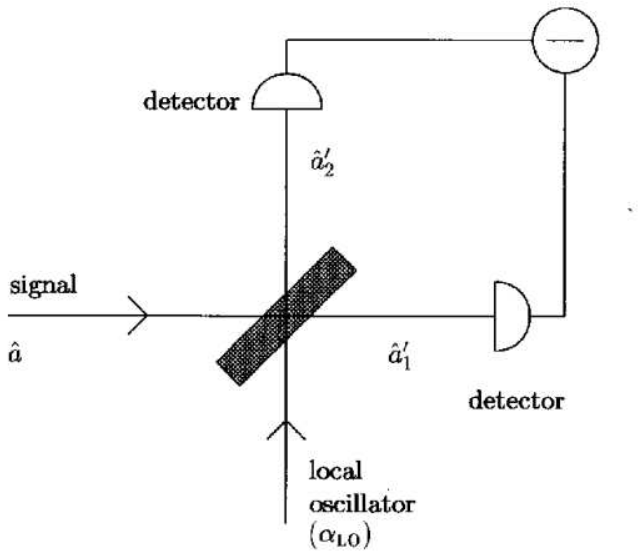


Figura 1.3: Diagramma di un ricevitore omodina [10]

Nella figura 1.3 viene rappresentato lo schema di una misura omodina. Il segnale $\hat{a}$ interferisce con un fascio coerente generato dal laser su un beam splitter $50{:}\,50$. Il fascio coerente, noto come oscillatore locale (LO) $\alpha_{LO} = 2^{-1/2}|\alpha_{LO}|e^{j\theta}$, fornisce la fase di riferimento θ per la misura della quadratura. Si assume che segnale e LO mantengano una relazione di fase costante, condizione

che si verifica se entrambi sono generati dallo stesso laser. Inoltre, è necessario che il LO sia più potente del segnale, in modo da fornire una fase di riferimento precisa e poter essere trattato in maniera classica, consentendo di trascurare le fluttuazioni quantistiche. Dopo la miscelazione ottica del segnale con l'oscillatore locale, i due raggi uscenti dal beam splitter $\hat{a}_1'$ e $\hat{a}_2'$ sono il risultato di una trasformazione lineare:

$$\begin{pmatrix} \hat{a}_1' \\ \hat{a}_2' \end{pmatrix} = BS \begin{pmatrix} \hat{a} \\ \alpha_{LO} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \hat{a} \\ \alpha_{LO} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{a} + \alpha_{LO} \\ \hat{a} - \alpha_{LO} \end{pmatrix} \tag{1.15}$$

I due rami in uscita dal BS $\hat{a}_1'$ e $\hat{a}_2'$ sono diretti verso due foto-rivelatori, dove vengono misurate elettronicamente le rispettive intensità di corrente I_1 e I_2. Queste quantità misurate sono proporzionali ai rispettivi numeri di fotoni dei fasci in uscita dal BS. Di conseguenza, l'operatore del numero di fotoni di $\hat{a}_1'$ e $\hat{a}_2'$ sono:

$$\hat{n}_1 = \hat{a}_1^\dagger \hat{a}_1 = \frac{1}{2}(\hat{a}^\dagger + \alpha_{LO}^*)(\hat{a} + \alpha_{LO}) = \frac{1}{2}(\hat{a}^\dagger \hat{a} + \hat{a}^\dagger \alpha_{LO} + \alpha_{LO}^* \hat{a} + \alpha_{LO}^* \alpha_{LO}) \tag{1.16}$$

$$\hat{n}_2 = \hat{a}_2^\dagger \hat{a}_2 = \frac{1}{2}(\hat{a}^\dagger - \alpha_{LO}^*)(\hat{a} - \alpha_{LO}) = \frac{1}{2}(\hat{a}^\dagger \hat{a} - \hat{a}^\dagger \alpha_{LO} - \alpha_{LO}^* \hat{a} + \alpha_{LO}^* \alpha_{LO}) \tag{1.17}$$

Successivamente, viene calcolata la differenza $I_{21} \equiv I_2 - I_1 \propto \hat{n}_2 - \hat{n}_1 = \hat{n}_{21}$:

$$\begin{aligned} \hat{n}_{21} = \hat{a}^\dagger \alpha_{LO} + \alpha_{LO}^* \hat{a} = \frac{|\alpha_{LO}|}{\sqrt{2}}(\hat{a}^\dagger e^{i\theta} + \hat{a} e^{-i\theta}) = \frac{|\alpha_{LO}|}{2}[(\hat{q} - i\hat{p})e^{i\theta} + (\hat{q} + i\hat{p})e^{-i\theta}] = \\ = \frac{|\alpha_{LO}|}{2}[\hat{q}(e^{i\theta} + e^{-i\theta}) + i\hat{p}(-e^{i\theta} + e^{-i\theta})] = |\alpha_{LO}|(\hat{q}\cos(\theta) + \hat{p}\sin(\theta)) \end{aligned} \tag{1.18}$$

Come si può evincere dall'equazione 1.18, attraverso una misura omodina, a seconda del valore dell'angolo θ, è possibile ottenere informazioni su $\hat{p}$ oppure $\hat{q}$. Infatti se $\theta = 0$, $\hat{n}_{21} = |\alpha_{LO}|\hat{q}$, mentre se $\theta = 90°$, $\hat{n}_{21} = |\alpha_{LO}|\hat{p}$.

Il rilevamento eterodina si basa su un metodo analogo, in cui il segnale viene suddiviso utilizzando un BS $50{:}\,50$ bilanciato aggiuntivo. Un ramo viene utilizzato per misurare $\hat{q}$, mentre l'altro, dopo uno aver effettuato uno sfasamento di $\pi/2$ all'oscillatore locale, per misurare $\hat{p}$.

A differenza del rivelatore omodina, che fornisce una sola e precisa misurazione, il rivelatore eterodina permette di misurare $\hat{p}$ e $\hat{q}$ simultaneamente. Tuttavia, le misurazioni effettuate sono soggette a un rumore aggiuntivo, dato dalla miscelazione del segnale con il vuoto quantistico all'ingresso del BS.

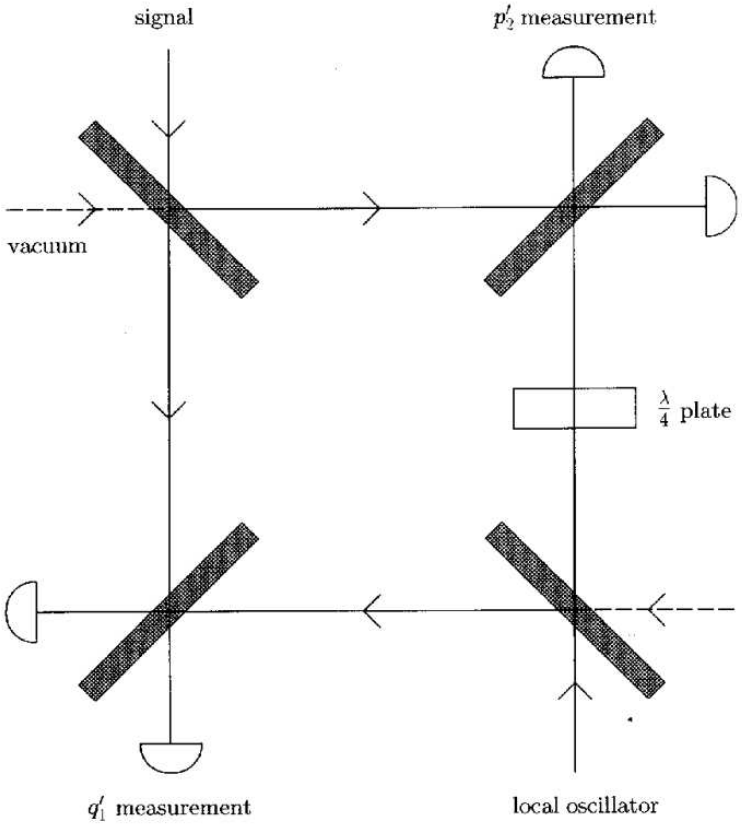


Figura 1.4: Schema di un rivelatore eterodina [10]

In conclusione il rilevamento eterodina fornisce informazioni su entrambe le quadrature del campo $\hat{p}$ e $\hat{q}$ del campo elettrico incidente. Questo campo corrisponde a uno degli stati coerenti preparati attraverso la modulazione di fase del fascio del laser. Il risultato di tale misurazione può essere rappresentato da un numero complesso β e associato ad un punto nello spazio delle fasi. Lo spazio delle fasi è uno spazio euclideo R^{2n}, dove n sono i gradi di libertà del sistema, ed è il prodotto dello spazio ordinario $\hat{q}$ e spazio di quantità di moto $\hat{p}$. [12]

1.2 Protocollo

In questo paragrafo verrà esposto il protocollo impiegato per un semi-device-indipendent CV-QRNG basato sull'eterodina adoperato per questa tesi.
Il protocollo si basa su due dispositivi principali: un dispositivo di preparazione parzialmente fidata e un blocco di misurazione non fidato. Il procedimento inizia con il blocco di preparazione, che, dopo aver ricevuto un input $x \in \{0,1,2,3\}$, emette un relativo stato quantico, il quale viene inviato al blocco di misurazione. Quest'ultimo blocco effettua una misura quantistica che genera un risultato $b \in \{0,1,2,3\}$.

Il blocco di misurazione è considerato come una scatola nera, di cui non si conoscono i dettagli interni del funzionamento. Si assume che il blocco di preparazione non sia correlato quantisticamente con l'attaccante, ma si assume che ci possa essere correlazione classica fra il sistema e l'ambiente. Inoltre, si suppone che l'energia degli stati preparati abbia un limite superiore:

$$\langle \hat{n} \rangle_{\rho_x} \leq \mu \tag{1.19}$$

dove $\hat{n}$ è l'operatore numero di fotoni, questa condizione è facilmente controllata e rispettata attraverso l'uso di misuratori di potenza ottica. Inoltre si assume che gli stati preparati siano indipendenti e identicamente distribuiti (i.i.d.), quindi gli stati hanno tutti la stessa distribuzione di probabilità e sono tutti statisticamente indipendenti.
Quando $\mu \leq 0.5$, questo limite superiore sull'energia comporta a un limite inferiore sulla sovrapposizione degli stati. Da questo limite è possibile seguire l'approccio descritto nell'articolo [13], che garantisce che la casualità generata sia autentica. Certificare la qualità della casualità generata richiede un'accurata stima dell'entropia. A tal fine, viene calcolato un limite sull'entropia dei dati di output utilizzando la distribuzione di probabilità condizionata, garantendo così che la casualità prodotta sia autentica e non alterata da fattori esterni. Questo processo è essenziale per prevenire che eventuali strategie classiche possano migliorare la prevedibilità dei risultati. Dopo aver certificato l'entropia, si passa all'estrazione di una stringa di bit casuali. Questo passaggio permette di ottenere un output che rispetti rigorosi standard di casualità e sicurezza, anche nel caso in cui i dispositivi utilizzati non siano completamente affidabili. L'obiettivo finale è assicurare che la stringa di bit sia priva di possibili correlazioni indesiderate, garantendo così un livello elevato di affidabilità anche in contesti potenzialmente non fidati.
Nell'implementazione sperimentale viene eseguita una modulazione Q-PSK, nella quale la sorgente è un laser a onda continua che genera i 4 stati coerenti: $|\psi_0\rangle = |\alpha\rangle e^{i\pi/4}, |\psi_1\rangle = |\alpha\rangle e^{i3\pi/4}, |\psi_2\rangle = |\alpha\rangle e^{i5\pi/4}, |\psi_3\rangle = |\alpha\rangle e^{i7\pi/4}$, con $\alpha = \sqrt{\mu} e^{i\phi}$, dove μ rappresenta la media del numero dei fotoni e ϕ la fase relativa tra il segnale e LO. A ciascun input x corrisponde uno specifico stato coerente, con l'input selezionato tramite un PRNG. Di conseguenza, la fase dello stato coerente verrà modulata per consentire l'invio dello stato corrispondente. Si assume che l'input x sia non correlato alle informazioni classiche disponibili e a chiunque conosca il funzionamento interno del dispositivo e indipendente dal dispositivo stesso.
Il ricevitore è modellizzato tramite una misura a valori operatoriali positivi (POVM) dell'eterodina. Lo spazio delle fasi può essere suddiviso in 4 regioni, una per ogni output possibile. Di conseguenza, dalla misurazione eterodina si ottiene un punto in un quadrante dello spazio delle fasi, a cui è associato l'output corrispondente. Questo punto appartiene a una delle quattro regioni possibili, determinando così uno dei quattro output b possibili.[6] [14]

Capitolo 2

Setup sperimentale

In questo capitolo verranno illustrate nel dettaglio le componenti del setup sperimentale del SDI-CV-QRNG basato sull'eterodina. Lo schema dell'apparato sperimentale è mostrato nella figura 2.1.

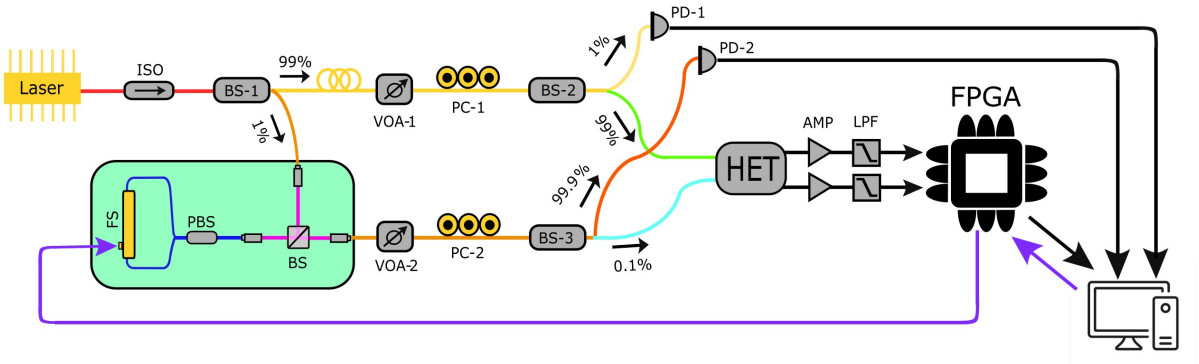


Figura 2.1: Schema del setup sperimentale del QRNG. Le sigle indicano: ISO = isolatore ottico, BS = beam-splitter, VOA = attenuattore ottico variabile, PC = controller di polarizzazione, PBS = polarizing beam splitter, HET = eterodina, AMP = amplificatore, LPF = filtro passa basso, FPGA = Field Programmable Gate Array.

Un laser a onda continua genera uno stato coerente alla lunghezza d'onda di 1550 nm. Il fascio di luce emesso dal laser passa attraverso un isolatore (ISO) e poi raggiunge un beam-splitter (BS) con rapporto di splitting $99{:}\,1$. L'isolatore ottico (ISO) è un componente ottico che consente la trasmissione della luce in una sola direzione. Viene generalmente utilizzato per prevenire feedback indesiderati in un oscillatore ottico e per evitare instabilità del laser a causa delle componenti riflesse. Il beam-splitter (BS), invece, è un dispositivo ottico che divide il fascio di luce incidente in 2 rami. A seconda del tipo di beam-splitter utilizzato, i due rami possono avere la stessa intensità (nel caso di un BS $50{:}\,50$) o intensità diverse. La potenza ottica e lo spettro

del laser è sensibile alle variazioni della temperatura ambientale, pertanto, viene controllato da un thermoelectric cooler (TEC) Controller. L'obiettivo principale non è solo mantenere la stessa potenza, ma garantire che il laser emetta in un solo modo, evitando la generazione di modi multipli. Questo dispositivo mantiene la temperatura del laser costante, mantenendola attorno ai $23°C$. Il TEC misura la temperatura del laser e la utilizza per regolare una cella Peltier in un sistema a retroazione. Attraverso un sistema di controllo PID (Proporzionale, Integrale, Derivativo), il TEC controlla in modo preciso e affidabile i sistemi di raffreddamento termoelettrici connessi, garantendo una stabilità termica ottimale per il laser.

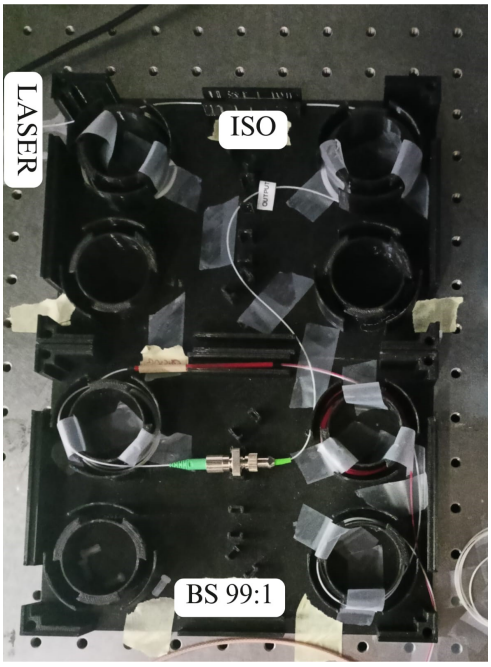


Figura 2.2: Immagine dell'isolatore, BS 99:1: il fascio di luce emesso dal laser passa attraverso un isolatore (ISO) e poi raggiunge un beam-splitter (BS) $99:1$

L'output 99% del BS viene utilizzato come oscillatore locale (LO). Successivamente, il fascio ottico passa attraverso un controller di polarizzazione (PC) e un attenuatore ottico variabile (VOA). Il controller di polarizzazione regola l'orientamento della polarizzazione del fascio, poiché l'eterodina è sensibile alla polarizzazione. Pertanto, è necessario garantire che il fascio incidente sia correttamente allineato in termini di polarizzazione. Invece l'attenuatore ottico variabile permette di modulare i livelli di potenza del segnale ottico. Successivamente, il fascio ottico incontra un ulteriore BS $99:1$. In questo secondo beam-splitter, la maggior parte dell'intensità ottica (ovvero il 99%) viene diretta verso l'eterodina (HET), il dispositivo principale per la rilevazione e l'analisi del segnale. Il restante 1% dell'intensità ottica è indirizzato verso un misuratore di potenza ottica (PM), utilizzato per misurare e monitorare la potenza del segnale

residuo. Questo controllo permette di verificare il rispetto del limite sull'energia imposto dal protocollo per garantire la vera casualità dei numeri generati.

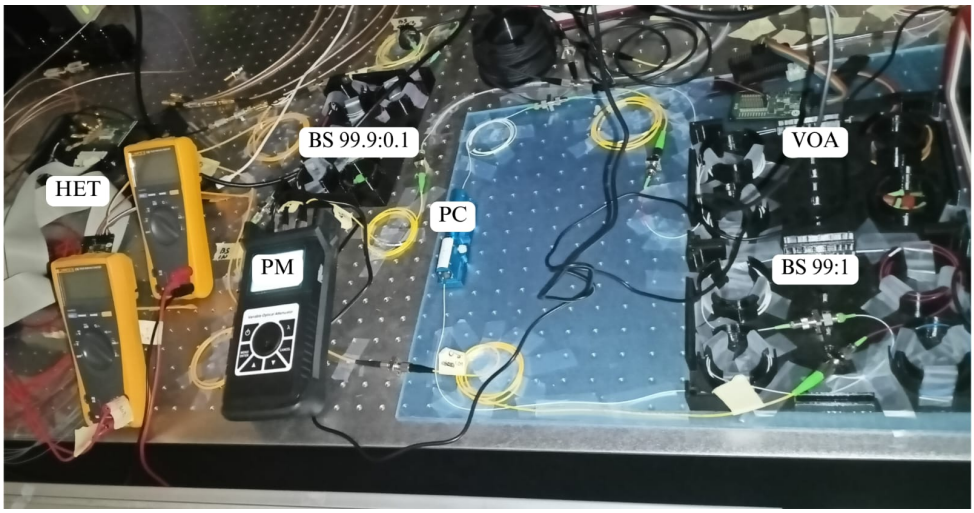


Figura 2.3: Immagine del percorso dell'oscillatore locale (LO): LO passa attraverso un controller di polarizzazione (PC) e un attenuatore ottico variabile (VOA), successivamente incontra un BS $99{:}\,1$. La maggior parte dell'intensità ottica (ovvero il 99%) viene diretta verso l'eterodina (HET), mentre il restante 1% dell'intensità ottica è indirizzato verso un misuratore di potenza ottica (PM).

La parte rimanente del BS a monte (1%) viene utilizzata come sorgente per il segnale di modulazione e passa attraverso un beam-splitter $50{:}\,50$ usato come circolatore. Successivamente, il segnale viene modulato sfruttando un loop di Sagnac composto da un polarizing beam splitter (PBS) e un modulatore di fase. In seguito, il segnale passa attraverso un controller di polarizzazione (PC), un attenuatore ottico variabile e un beam-splitter $99.9{:}\,0.1$, di cui la parte più intensa (99.9%) è collegata a un misuratore di potenza, mentre l'altra (0.1%) è diretta all'eterodina. L'insieme di beam-splitter, VOA e misuratore di potenza sono utilizzati per monitorare e controllare l'attenuazione del segnale, che può essere impostato arbitrariamente anche fino al livello di singolo fotone.

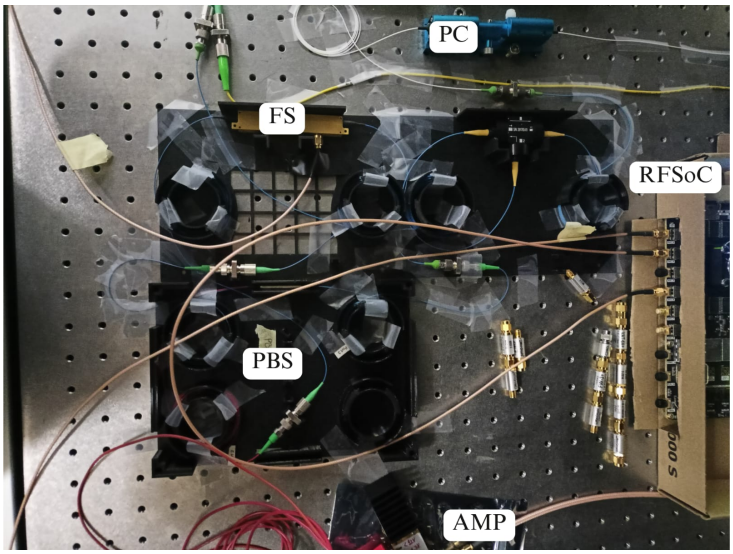


Figura 2.4: Immagine del percorso di modulazione del segnale: la forma d'onda generata dal RFSoC viene amplificata e inviata al modulatore di fase (FS). Successivamente, il segnale viene modulato sfruttando un loop di Sagnac composto da un polarizing beam splitter (PBS) e un modulatore di fase (FS).

L'eterodina usata in questo esperimento è sensibile alla polarizzazione, pertanto sia il segnale e sia l'oscillatore locale devono avere la polarizzazione allineata. In caso contrario, i due modi in entrata all'eterodina non interferiscono massimamente e si verifica una perdita significativa dell'intensità ottica. In seguito, il segnale elettrico in uscita dall'eterodina viene amplificato per essere letto da un Field Programmable Gate Array (FPGA) Zynq Ultrascale+ RFSoC XCZU48DR-2FFVG1517E con dei convertitori analogico-digitale (ADC) integrati, con cui si riesce a elaborare il segnale letto tramite computer.

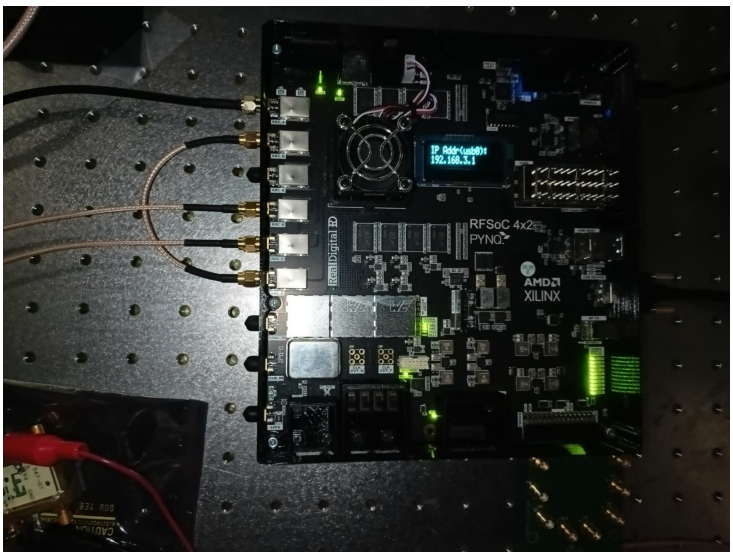


Figura 2.5: Immagine del RFSoC (Radio Frequency System-on-Chip), utilizzato per generare le forme d'onda per la modulazione del segnale e per acquisire il segnale dall'eterodina.

Il RFSoC non solo gestisce l'elaborazione del segnale, ma viene usato anche per modulare la fase dei segnali ottici necessari per il protocollo. La selezione della forma d'onda da inviare nel setup è effettuata tramite computer, mentre il RFSoC utilizza un convertitore digitale-analogico (DAC) integrato con una frequenza di campionamento di 4.0 GHz e una tensione in uscita di 210 mV. Tuttavia, il modulatore di fase richiede una tensione massima per la modulazione di 7 V, quindi è stato necessario inserire un pre-amplificatore e un amplificatore in cascata per aumentare la tensione in uscita dal RFSoC.
Inoltre le risorse del RFSoC, come il processore interno, possono essere controllate dal framework PYNQ. Utilizzando Jupyter Notebook, è possibile interagire con il dispositivo e visualizzare o analizzare le forme d'onda tramite un'interfaccia basata su Python eseguita su un PC collegato. Un Notebook Jupyter può eseguire codice e processare i dati acquisiti dal RFSoC in tempo reale, facilitando l'analisi e il controllo del sistema.
Inizialmente, nel set-up sperimentale, indicato nella figura 2.1, al posto del RFSoC era presente un generatore di forme d'onda Siglent $SDG6032X$, con 2.4 GSa/s di massima frequenza di campionamento per modulare il segnale e in ricezione un oscilloscopio Tektronix $DPO70404C$. L'oscilloscopio Tektronix, invece, acquisiva i dati, che venivano successivamente salvati in file per essere elaborati su un computer. Similmente al RFSoC, il generatore Siglent si collegava online ed era possibile selezionare dal computer la forma d'onda da inviare. Tuttavia, in seguito a un guasto del generatore Sigent, si è optato per l'uso sia in ricezione, sia per la generazione delle forme d'onda del RFSoC. In questo modo, tutte le funzioni di modulazione e acquisizione sono state centralizzate nel RFSoC, semplificando il setup e migliorando l'efficienza del processo sperimentale.

Capitolo 3

Metodologia

In questo capitolo verrà esposto il procedimento per arrivare a un'ottimizzazione della generazione di stati per un generatore di numeri quantistico. Nella prima parte di questo esperimento, il RFSoC viene utilizzato come un generatore di forme d'onda per produrre una precisa forma d'onda arbitraria, illustrata nella figura 3.1.

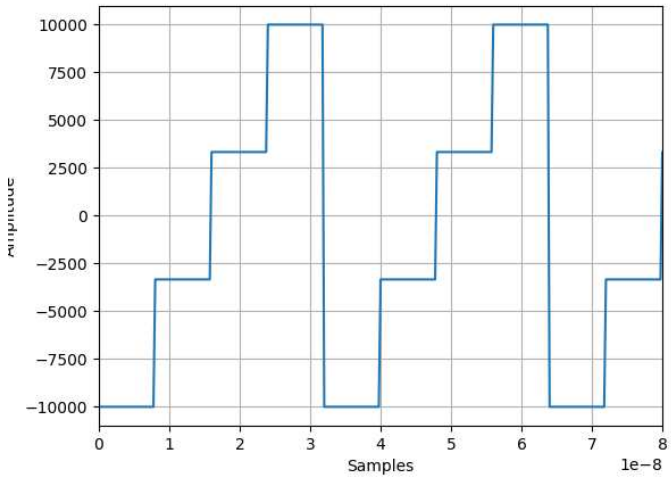


Figura 3.1: Forma d'onda arbitraria a scalini.

Per implementare il protocollo di generazione di numeri casuali scelto è necessario generare una costellazione Q-PSK, corrispondente a quattro stati coerenti con stessa energia media, ma con quattro fasi diverse: $|\psi_0\rangle = |\alpha\rangle e^{i\pi/4}, |\psi_1\rangle = |\alpha\rangle e^{i3\pi/4}, |\psi_2\rangle = |\alpha\rangle e^{i5\pi/4}, |\psi_3\rangle = |\alpha\rangle e^{i7\pi/4}$, con $\alpha = \sqrt{\mu} e^{i\phi}$, dove μ rappresenta la media del numero dei fotoni e ϕ la fase relativa tra il segnale e LO. Per ottenere la modulazione di fase richiesta, come mostrato nel capitolo precedente, si utilizza un modulatore di fase polarization-insensitive, all'interno di un interferometro di Sagnac. Tale modulatore permette di aggiungere una fase addizionale alla luce che lo attraversa,

come mostrato nell'equazione 1.2, in funzione di un voltaggio elettrico fornito al modulatore stesso. In particolare, per generare i quattro stati richiesti si può utilizzare un segnale a scalini come mostrato in Figura 3.1. L'ampiezza e la durata dei singoli scalini di tale forma d'onda, permettono di variare le modulazioni di fase impresse al segnale ottico. Questo segnale ottico modulato viene poi diretto al ricevitore ottico eterodino, che effettua la misura e la conversione da segnale ottico a segnale elettrico. Infine, il segnale elettrico generato dall'eterodina viene digitalizzato dal RFSoC. Attraverso questo dispositivo, è possibile leggere i due canali dell'eterodina e ricavare la costellazione dei simboli generati. L'obiettivo di questa tesi è quello di ottimizzare la generazione delle forme d'onda che pilotano il modulatore di fase, in modo da ottimizzare i simboli della costellazione Q-PSK ricevuti per ottenere una costellazione equiangolare con separazione di $90°$ tra i simboli. A causa delle non idealità presenti nel set-up sperimentale, determinare i valori ottimali della forma d'onda in maniera analitica risulta complesso. Si è scelto di sviluppare un algoritmo che, iterativamente, converga al valore ottimale, misurando ad ogni step la distanza dalla configurazione target. L'algoritmo aggiorna iterativamente il valore dei parametri delle forme d'onda recuperando informazioni sull'angolo dei simboli nella costellazione dal segnale ricevuto dall'eterodina.

3.1 Classe per la generazione delle forme d'onda

In questo paragrafo viene spiegato il processo di generazione dei diversi stati quantistici e viene descritto il codice che è stato sviluppato per comunicare e programmare sia il RFSoC che il generatore Siglent utilizzati per la generazione delle forme d'onda.

3.1.1 Siglent SDG6032X

Il generatore Siglent può essere utilizzato come generatore di forme d'onda arbitrarie. In questa modalità è possibile inviare allo strumento una forma d'onda arbitraria che verrà replicata. Per automatizzare questo processo, è possibile interagire con il dispositivo in remoto utilizzando PyVISA. Attraverso uno script Python, si possono inviare comandi per selezionare e caricare una forma d'onda, impostare l'impedenza desiderata o attivare e disattivare i canali disponibili. Di seguito è riportato un esempio di stringa utilizzata per generare il messaggio con la corretta formattazione per sintetizzare una funzione sinusoidale:

```
wave_sine = f'C{chan}:BSWV WVTP,SINE,FRQ,{freq},AMP,{amp},OFST,{ofst}'
```

La variabile `chan` seleziona il canale del dispositivo (1 o 2), mentre `freq` indica la frequenza, con un massimo di 500 MHz. La variabile `amp` rappresenta la tensione picco-picco della funzione sinusoidale, e `ofst` indica l'offset. Per generare una funzione arbitraria a scalini, si utilizza

tuttavia un'altra modalità. In questo caso, viene creato un array contenente i valori di tensione corrispondenti ai livelli della forma d'onda desiderata. Questi valori sono poi quantizzati, tenendo conto della soglia di saturazione del generatore, poiché il dispositivo ha limiti di tensione in uscita. L'array di valori decimali viene convertito in formato esadecimale, con convenzione little-endian, e infine codificato in una stringa ASCII. I dati vengono quindi codificati in "latin1" e inviati al dispositivo. Per configurare la frequenza, l'ampiezza picco-picco e l'offset, viene inviata un'altra stringa di comando, di seguito è riportato un esempio di tale stringa con la corretta formattazione:

```
wave_data = f'C{chan}:BSWV FRQ,{freq},AMP,{amp},OFST,{ofst}'
```

La variabile `chan` seleziona il canale del dispositivo (1 o 2), mentre `freq` indica la frequenza, mentre la variabile `amp` rappresenta la tensione picco-picco della funzione sinusoidale, e `ofst` indica l'offset. Per agevolare l'utilizzo di questo codice, queste funzionalità di controllo sono state integrate all'intero di una classe Python che, attraverso il suo costruttore, stabilisce la connessione con il generatore. Questa classe include metodi per inviare sia forme d'onda classiche sia arbitrarie, permettendo la selezione del canale, della frequenza, dell'ampiezza picco-picco e dell'offset.

3.1.2 RFSoC

La scheda RFSoC possiede un generatore di forme d'onda DAC (Digital-to-Analog Converter), che consente la sintetizzazione di forme d'onda arbitrarie, con banda passante superiore al GHz. Quando la scheda è collegata alla rete, è possibile interagire con essa tramite Jupyter Lab IDE. Similmente al generatore Siglent, la generazione della forma d'onda arbitraria avviene inviando un array con il valore dell'ampiezza dell'onda specificato per ogni punto di sampling. Dato che il software sulla shceda RFSoc è basato su Python è stata utilizzata la libreria Numpy, e poi sono state utilizzate le API dell'RFSoC per caricare l'array sulla scheda.
Per esempio, per inviare la funzione seno, si può procedere come segue:

```
from rfsoc_mts import mtsOverlay
import numpy as np

#Per accedere alla memoria DAC
ol = mtsOverlay('mts.bit')

# Creare l'array con la forma d'onda
DAC_SR = 4.0E9  # Frequenza di campionamento dei DACs è 4.0 GHz
DAC_Amplitude = 16383.0  # 14bit DAC +16383/-16384
```

```
Fc = 500.0E6 # Frequenza della forma d'onda 500.0 MHz
X_axis = (1/DAC_SR) * np.arange(0, ol.dac_player.shape[0])
DAC_sinewave = DAC_Amplitude * np.sin(2 * np.pi * Fc * X_axis)

#Inviare la forma d'onda al DAC
ol.dac_player[:] = np.int16(DAC_sinewave)
```

Per inviare una funzione arbitraria a scalini utilizzando l'RFSoC, è sufficiente selezionare i 4 livelli desiderati e ripeterli in un array di 65536 elementi. I livelli vengono ripetuti all'interno dell'array, con la frequenza di ciascun livello che determina la frequenza della forma d'onda generata. Successivamente, si carica l'array nella memoria del DAC del RFSoC, permettendo la generazione della forma d'onda a scalini desiderata. Le frequenze disponibili sono quantizzate e comprendono: 125 MHz, 250MHz, 500MHz, 1 GHz. Una volta inviato il segnale e una volta iniziata la generazione, questa avviene in maniera periodica, ripetendo ciclicamente i dati caricati in memoria. Nella situazione in cui sia necessario interrompere il segnale, bisogna sovrascrivere nella memoria del DAC un array di 0. La classe Python sviluppata per interfacciarsi con l'RFSoC è strutturata in modo simile a quella utilizzata per il generatore Siglent. Il costruttore della classe inizializza l'overlay MTS del RFSoC e include varie funzioni per inviare sia forme d'onda sinusoidali sia forme d'onda a livelli arbitrari.

3.2 Classe per la ricezione

In questo paragrafo viene illustrata la procedura di acquisizione dei dati dall'eterodina attraverso l'RFSoC. Il segnale ottico dopo la modulazione attraversa l'intero il set-up sperimentale, giunge all'eterodina qui viene rivelato, convertito in segnale elettrico e poi acquisito dalla scheda RFSoC. Quest'ultima si occupa della digitalizzazione tramite l' ADC interno. A questo dispositivo sono collegati 3 canali in ricezione. Nella figura 3.2 è mostrato un esempio di campioni catturati dei 3 canali, dove il canale 0 indica la maschera usata, mentre il canale 2 e 3 sono dedicati ai segnali provenienti dall'eterodina, che verranno riferiti rispettivamente con `"sig_mask"`, `"sig1"` e `"sig2"`. Il termine maschera indica il segnale elettrico generato dalla scheda RFSoC, che viene amplificato, utilizzato per modulare il segnale ottico e successivamente acquisito dalla stessa scheda RFSoC. Per quanto concerne i segnali provenienti dall'eterodina, è stato illustrato nel capitolo introduttivo che la misura eterodina fornisce informazioni su entrambe le quadrature del campo $\hat{p}$ e $\hat{q}$ del campo elettromagnetico incidente. Di conseguenza, i segnali del canale 2 e 3 contengono informazioni relative a una specifica quadratura del campo elettromagnetico: uno è associato alla quadratura $\hat{p}$, mentre l'altro è riferito alla quadratura $\hat{q}$.

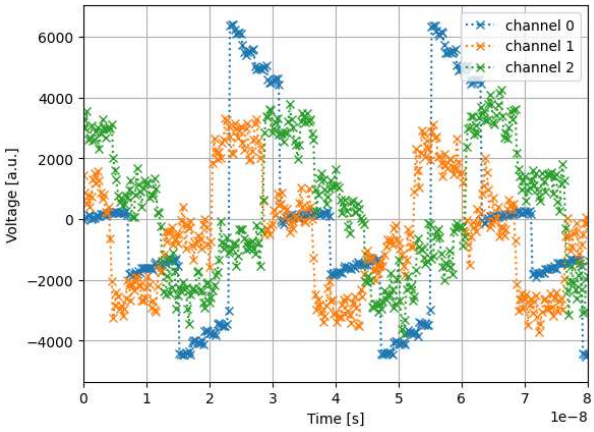


Figura 3.2: i 3 canali ADC in ricezione del RFSoC, il channel 0 è il segnale della maschera, mentre channel 1 e channel 2 sono i segnali dell'eterodina.

Una volta acquisiti i segnali elettrici, è necessario procedere al campionamento. Tuttavia i segnali `sig1` e `sig2`, rispetto alla maschera, percorrono una distanza maggiore prima di giungere alla scheda del RFSoC. Di conseguenza, al momento della cattura dei segnali, la maschera non è allineata temporalmente ai segnali elettrici dell'eterodina. Pertanto, è necessario effettuare uno shift dei dati per compensare il ritardo introdotto dalla diversa distanza percorsa dai segnali, ciò viene effettuato tramite il metodo "roll" dell'array in Python. In seguito, si esegue la correlazione tra la maschera e il segnale inviato dal RFSoC per identificare il punto massimo, che viene utilizzato come istante iniziale per il campionamento. A partire da questo punto massimo, si seleziona sistematicamente il punto medio del periodo del simbolo. Durante il processo vengono inviati quattro simboli e acquisiti due canali dall'eterodina. Dopo il campionamento, la costellazione risultante è rappresentata come segue:

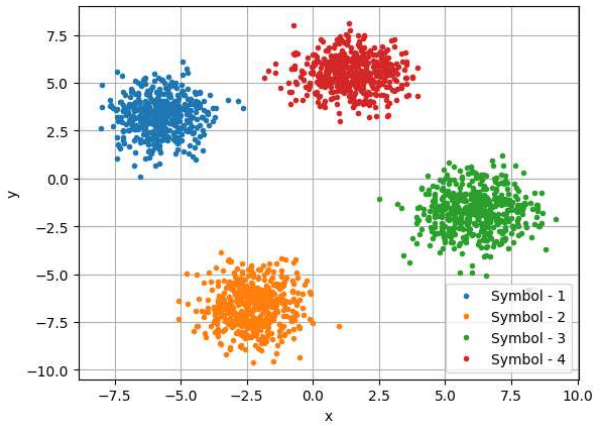


Figura 3.3: La costellazione in ricezione non è ottimizzata, visto che i cluster della costellazione sono ben distinti, senza sovrapposizioni. Tuttavia, gli angoli relativi tra i centroidi dei cluster non risultano tutti uguali a 90°.

Ogni cluster corrisponde a un simbolo inviato. Questi cluster dovrebbero essere idealmente disposti in maniera equiangolare a una distanza relativa di $90°$ dai simboli adiacenti. Per raggiungere questo obiettivo, si determina inizialmente il centroide di ciascun cluster e si procede al calcolo del suo angolo in questa specifica acquisizione. Successivamente, si determinano gli angoli relativi tra i punti medi. Nella figura 3.3 sono riportati gli angoli relativi risultanti: $100.7°$, $93.3°$, $91.0°$ e $75.0°$.
Come si può notare, sia dai valori degli angoli relativi che visualmente dalla figura, questa particolare configurazione risulta lontana dalla configurazione ideale di una costellazione equiangolare. Per ottimizzare questa configurazione è stato sviluppato un sistema automatico con feedback.

3.3 Algoritmo di ottimizzazione

In questo paragrafo viene illustrato l'algoritmo di ottimizzazione, utilizzato per ottimizzare la forma d'onda e per ottenere la costellazione target. Una volta acquisiti gli angoli relativi viene calcolato il loro errore, definito come la differenza tra l'angolo relativo ideale di $90°$ e la media sugli angoli relativi acquisiti.

L'errore, denominato `"dphi"`, viene memorizzato per ciascun livello, associandolo ad uno specifico angolo relativo. A causa delle incertezze intrinseche al set-up sperimentale, in particolare dovute all'eterodina, si riscontra una notevole variabilità negli angoli misurati. Per mitigare questo effetto e ottenere risultati più accurati, la medesima forma d'onda viene trasmessa e acquisita ripetutamente, con il numero di acquisizioni definito dalla variabile `num_acq`. I risultati vengono quindi mediati per aumentare la precisione delle misurazioni. Nelle fasi iniziali dell'ottimizzazione, quando i valori di `dphi` possono essere considerevoli, si mantiene un `num_acq` basso per consentire iterazioni rapide. L'algoritmo si concentra sul livello caratterizzato dal `dphi` più elevato, adottando due possibili strategie di ottimizzazione:

- Modifica con passo fisso: aggiunta o sottrazione di un valore costante;
- Modifica con passo variabile: aggiunta o sottrazione di un valore che varia dinamicamente.

Il processo di ottimizzazione prevede la modifica dei livelli della forma d'onda in base ai valori di `dphi`. Un `dphi` positivo comporta un incremento all'iterazione successiva del livello corrispondente, mentre un `dphi` negativo ne determina una diminuzione. È importante notare che il primo e l'ultimo livello della forma d'onda possono raggiungere una condizione di saturazione, data la limitazione del range di uscita del RFSoC. In tali circostanze, l'algoritmo genera

un avviso e cessa di intervenire sul livello in questione. Qualora entrambi i livelli raggiungessero la saturazione, l'algoritmo potrebbe non convergere verso una soluzione ottimale.

L'algoritmo considera di aver raggiunto una configurazione ottimale e termina il processo di ottimizzazione quando il `dphi` più elevato risulta inferiore alla condizione d'uscita prestabilita. È fondamentale non impostare una soglia di uscita eccessivamente bassa, poiché l'incertezza intrinseca nella misura degli angoli potrebbe impedire all'algoritmo di convergere in tempi ragionevoli. Infatti, tra due acquisizioni successive del RFSoC, effettuate con la medesima forma d'onda, l'angolo può variare fino a $10°$ a causa delle non idealità del set-up sperimentale, alla statistica finita e all'elevato rumore nel sistema. Ciò sottolinea l'importanza di acquisire un numero adeguato di costellazioni e di calcolare la media degli angoli relativi per ottenere risultati più affidabili. Inoltre, una soglia di uscita troppo stringente può indurre oscillazioni nell'algoritmo, causando un'alternanza tra stati prossimi alla conclusione e stati significativamente distanti.

Per ottimizzare l'efficienza del processo, il numero di acquisizioni (`num_acq`) viene inizialmente impostato a 2, considerando che il `dphi` è elevato e non è richiesta un'alta precisione. Tuttavia, al diminuire del `dphi`, il numero di acquisizioni viene incrementato per aumentare la precisione del sistema. Specificamente:

- Quando `dphi` $< 20°$: `num_acq` = 6
- Quando `dphi` $< 10°$: `num_acq` = 12
- Quando `dphi` $< 5°$: `num_acq` = 18

Questa strategia adattiva consente di affinare gradualmente la precisione del sistema, modulando il numero di acquisizioni in funzione dell'errore misurato, garantendo così un equilibrio ottimale tra velocità di convergenza e accuratezza dei risultati.

3.3.1 Processo iterativo di ottimizzazione

L'algoritmo opera secondo il seguente schema iterativo:

1. Invio di una forma d'onda a scalini iniziale.
2. Acquisizione multipla della risposta del sistema, con un numero di acquisizioni variabile in base all'errore sugli angoli relativi.
3. Calcolo della media dei risultati ottenuti dalle diverse acquisizioni.
4. Decisione sulla modifica di un livello della forma d'onda basata sull'errore calcolato.
5. Invio della forma d'onda aggiornata al set-up sperimentale.

6. Ripetizione del processo fino al raggiungimento di un errore sugli angoli relativi accettabile.

Questo approccio iterativo consente all'algoritmo di adattarsi dinamicamente alle caratteristiche del sistema, bilanciando efficacemente velocità di convergenza e precisione del risultato finale.

3.3.2 Ottimizzazione con passo fisso

Questo approccio utilizza un passo costante per modificare il valore dei livelli. Sebbene un passo più ampio possa essere vantaggioso nelle fasi iniziali del processo, potrebbe risultare inefficace successivamente, impedendo all'algoritmo di convergere verso una soluzione più precisa. Questo fenomeno si verifica poiché l'algoritmo potrebbe oscillare tra due soli livelli, senza esplorare efficacemente altre configurazioni.

Per affrontare questa problematica, è stato implementato un approccio adattivo:

- Inizialmente, si utilizza un passo `dvolt` pari a 0,05.
- Quando `dphi` scende sotto i $10°$, il passo viene ridotto a 0,01 per migliorare la precisione.

Nel seguito, ci si riferirà a questo algoritmo con l'abbreviazione "df" (delta fisso).

3.3.3 Ottimizzazione con passo variabile

In questa configurazione, il passo di ottimizzazione varia ad ogni iterazione. Si seleziona un `dvolt` che viene moltiplicato per l'errore corrente e diviso per una variabile di scala `dv_rescale`. La scelta di `dv_rescale` è critica: un valore troppo piccolo potrebbe portare a modifiche irrilevanti della forma d'onda, prolungando il tempo di convergenza.

Per ottimizzare il processo, `dv_rescale` viene adattato dinamicamente in base al valore massimo di `dphi`:

- `dphi` $< 20°$: `dv_rescale` = 40,0
- `dphi` $< 10°$: `dv_rescale` = 50,0
- `dphi` $< 5°$: `dv_rescale` = 80,0

Questo algoritmo sarà indicato con l'abbreviazione "dv" (delta variabile).

Capitolo 4

Risultati

In questo capitolo verrà mostrato il comportamento dell'algoritmo e i relativi risultati ottenuti ai vari test.
L'obiettivo principale di questo studio è stato l'ottimizzazione della costellazione dei simboli generati tramite modulazione di fase, al fine di ottenere una costellazione in cui i simboli siano equidistanti di $90°$. Al fine di conseguire questo risultato e a causa delle imperfezioni nel set-up sperimentale, si è proceduto allo sviluppo di un algoritmo per ottimizzare i livelli della forma d'onda e avvicinarsi alla costellazione ideale. L'algoritmo si basa su un processo iterativo che valuta l'errore angolare e adatta i livelli della forma d'onda per minimizzare l'errore fino a raggiungere la configurazione desiderata. Nella presente analisi sono stati utilizzati due algoritmi distinti per l'ottimizzazione, ciascuno caratterizzato da un diverso approccio al passo di aggiornamento della forma d'onda. Il primo algoritmo utilizza un passo fisso, indicato con la notazione `df`. Il secondo algoritmo adotta un passo variabile, denotato con `dv`. Il secondo algoritmo consente una variazione dinamica della dimensione del passo di aggiornamento in funzione delle errore. Entrambi gli approcci hanno i loro vantaggi e limitazioni, e questo capitolo fornirà dettagli specifici sulle prestazioni raggiunte, sull'efficacia dell'algoritmi di ottimizzazione e sulle sfide incontrate durante l'esperimento.
Per verificare la validità dell'algoritmo, si è valutato il suo funzionamento partendo da condizioni iniziali diverse, ovvero utilizzando forme d'onda con livelli diversi e osservando il processo di convergenza. Inoltre, per accertarsi dell'assenza di errori latenti, si è ricostruita la costellazione in ricezione per valutare se l'algoritmo fosse effettivamente in grado di equi distanziare i cluster della costellazione. Nella seguente tabella vengono sintetizzati i valori utilizzati come parametri iniziali per i vari test effettuati

Punti iniziali	livello 1	livello 2	livello 3	livello 4
prova con i livelli simmetrici (test 1)	0.7	0.35	-0.35	-0.7
prova con i livelli asimmetrici (test 2)	0.8	0.4	-0.3	-0.6
prova con i livelli prossimi alla saturazione (test 3)	0.7	0.6	-0.5	-0.6
prova con i livelli prossimi allo zero (test 4)	0.2	0	-0.1	-0.4

Tabella 4.1: Tabella dei punti iniziali

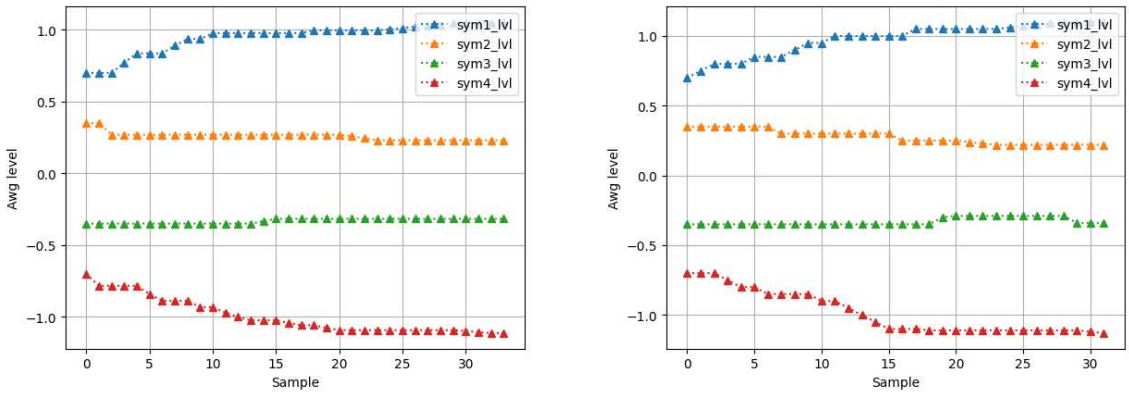
In tutti i test vi sono alcune parametri comuni:

1. Condizione d'uscita posta a 3.5 °;
2. Livello di saturazione posto a 1.3;
3. `dvolt` $= 0.05$ e `dv_rescale` $= 30.$;

La condizione d'uscita è posta a 3.5 ° rappresenta un compromesso adeguato tra una soglia di uscita non eccessivamente bassa, che potrebbe compromettere la convergenza dell'algoritmo a causa dell'incertezza intrinseca nella misura degli angoli, e un margine di errore accettabile per ottimizzare i cluster della costellazione. Il livello di saturazione posto a 1.3 è necessario per i limiti fisici imposti dalle componenti usate, in particolare il convertitore digitale-analogico (DAC) del RFSoC ha una soglia nel suo range d'uscita per le forme d'onda generate. Questo implica che, al di sopra di tale soglia, anche se venisse richiesto dall'algoritmo un valore di tensione più elevato, l'RFSoC non sarebbe in grado di generarlo. Senza tale limite, l'algoritmo potrebbe bloccarsi su quel livello saturato, modificandolo iterazione dopo iterazione senza registrare effettivi miglioramenti nella costellazione. Il parametro `dvolt` $= 0.05$ è stato selezionato per evitare di avere un passo troppo ampio, in quanto possa essere vantaggioso nelle prime iterazioni del processo, ma potrebbe risultare inefficace successivamente, impedendo all'algoritmo di convergere verso una soluzione più precisa. Invece, il parametro `dv_rescale` $= 30.$ è stato scelto per garantire che le modifiche alla forma d'onda non siano irrilevanti ed evitino di prolungare il tempo di convergenza.

4.1 Acquisizione con i livelli simmetrici

In questa sezione saranno illustrate le prestazioni dell'algoritmo utilizzando una configurazione di partenza caratterizzata da livelli simmetrici. In particolare, verranno analizzati i risultati ottenuti con una condizione iniziale definita dai valori 0.7, 0.35, -0.35 e -0.7. I dati raccolti forniranno indicazioni dettagliate sul comportamento dell'algoritmo in presenza di livelli simmetrici e contribuiranno a verificare la sua capacità di raggiungere e mantenere la configurazione ideale della costellazione.



(a) L'andamento dei livelli della forma d'onda usando un passo variabile (dv)

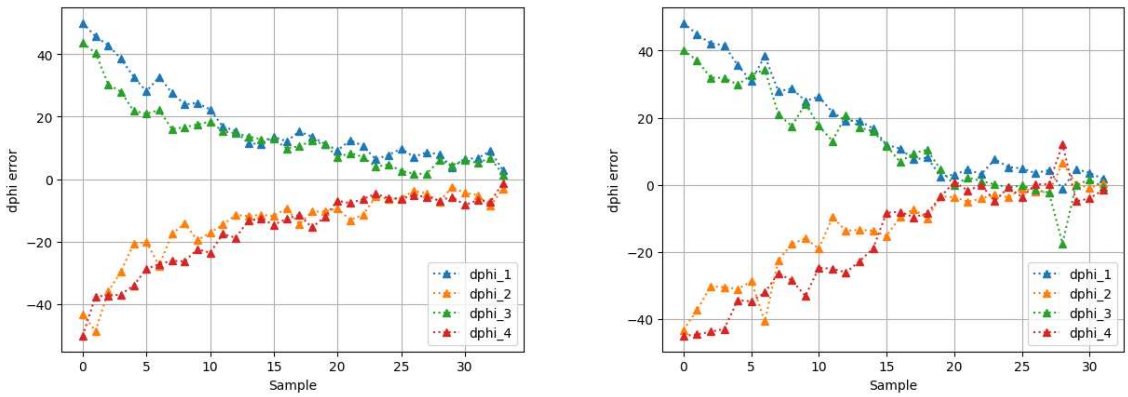
(b) L'andamento dei livelli della forma d'onda con usando un passo fisso (df)

Figura 4.1: Modifiche ai livelli della forma d'onda apportate dall'algoritmo. Le variazioni illustrate mostrano l'impatto dell'algoritmo sui diversi livelli della forma d'onda, evidenziando il processo di generazione del segnale.

Nelle figure 4.1a e 4.1b sono presentati gli andamenti dei livelli settati dal DAC in funzione del numero delle iterazioni dell'algoritmo utilizzato. Nella seguente tabella 4.2 sono illustrati i livelli della forma d'onda raggiunti degli algoritmi nel test 1.

Livelli raggiunti da dv1	1.05	0.23	-0.32	-1.11
Livelli raggiunti da df1	1.10	0.22	-0.34	-1.13

Tabella 4.2: Livelli raggiunti dagli algoritmi nel test 1.



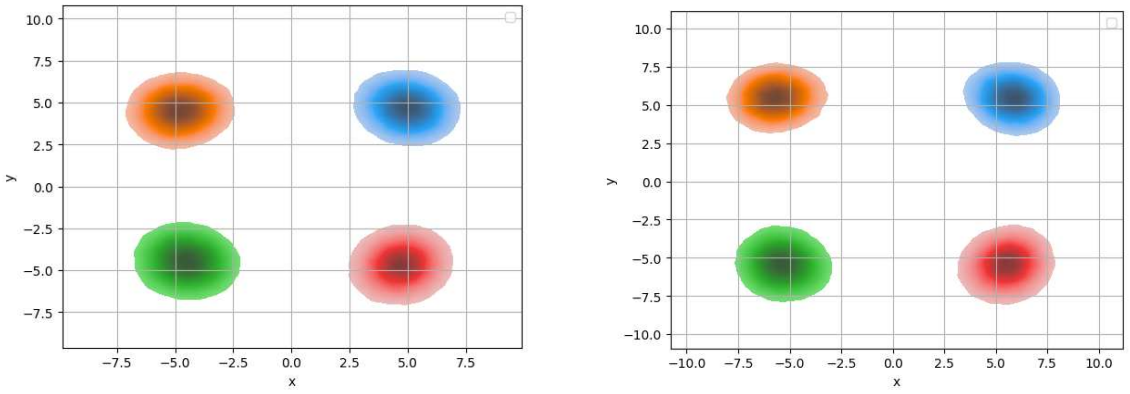
(a) L'andamento dell'errore con passo variabile nel test con i livelli simmetrici (dv1)

(b) L'andamento dell'errore con passo fisso nel test con i livelli simmetrici (df1)

Figura 4.2: Andamento dell'errore con gli algoritmi durante il test e si osserva come l'errore venga progressivamente minimizzato.

Nelle figure 4.2a e 4.2b sono presentate le convergenze degli errori rispetto agli angoli target degli algoritmi. Entrambi gli algoritmi presentano un andamento simile, tuttavia l'algorit-

mo df ha raggiunto una configurazione ottimale con un numero minore di iterazioni rispetto all'algoritmo dv.



(a) Costellazione della configurazione d'uscita (dv1) (b) Costellazione della configurazione d'uscita (df1)

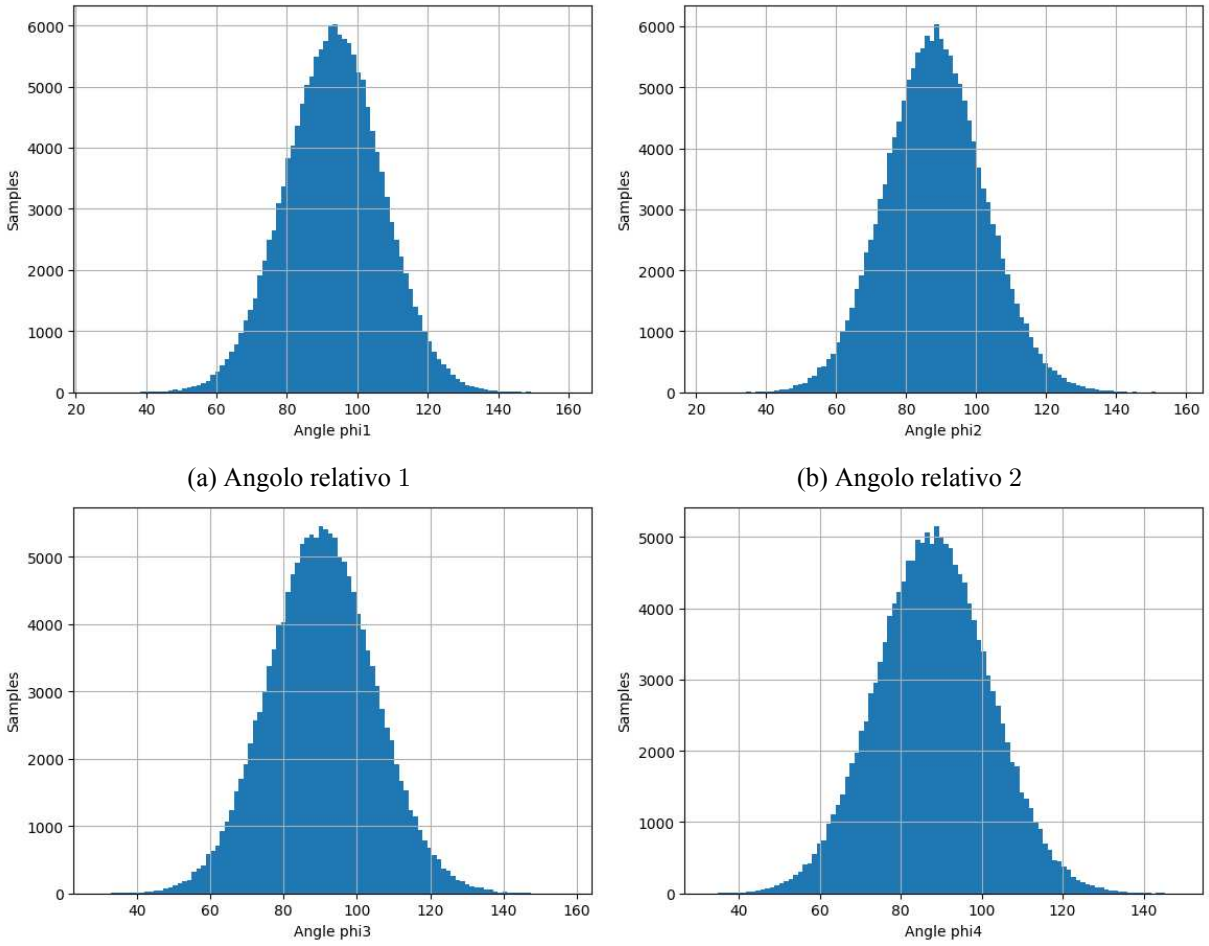
Figura 4.3: Costellazioni ottenute dalla media di 152.400 simboli, calcolata a partire dalla forma d'onda registrata alla condizione finale di uscita del rispettivo algoritmo. Questi dati illustrano la stabilità e l'affidabilità dei simboli generati, in quanto i cluster non sono sovrapposti.

Le costellazioni ricostruite al ricevitore indicate nelle figure 4.3a e 4.3b sono ottenute dalla media di 152.400 simboli, calcolata a partire dalla forma d'onda trovata alla condizione finale di uscita del rispettivo algoritmo. La media degli angoli relativi di tutti i simboli acquisiti durante questa acquisizione estesa è risultata essere:

Media degli angoli relativi di dv1	93.30	88.24	90.41	88.05
Media degli angoli relativi di df1	92.90	89.56	90.02	87.52

Tabella 4.3: Media degli angoli relativi della lunga acquisizione della prova 1. La lunga acquisizione è composta da 300 misurazioni effettuate dal RFSoC, corrispondenti a un totale di 152.400 simboli acquisiti. Questi risultati mostrano come l'algoritmo riesca a raggiungere la configurazione ottimale.

Come si può vedere dalla tabella 4.3, entrambi gli algoritmi hanno dimostrato un'efficace ottimizzazione della costellazione, con gli angoli relativi che si discostano dall'angolo target di massimo 3.30° nel primo algoritmo e con un errore massimo di 2.90° nel secondo. Le figure 4.4 e 4.5 mostrano che la distribuzione di tutti gli angoli relativi, presi durante una lunga acquisizione seguono una distribuzione gaussiana centrata attorno all'angolo 90°:



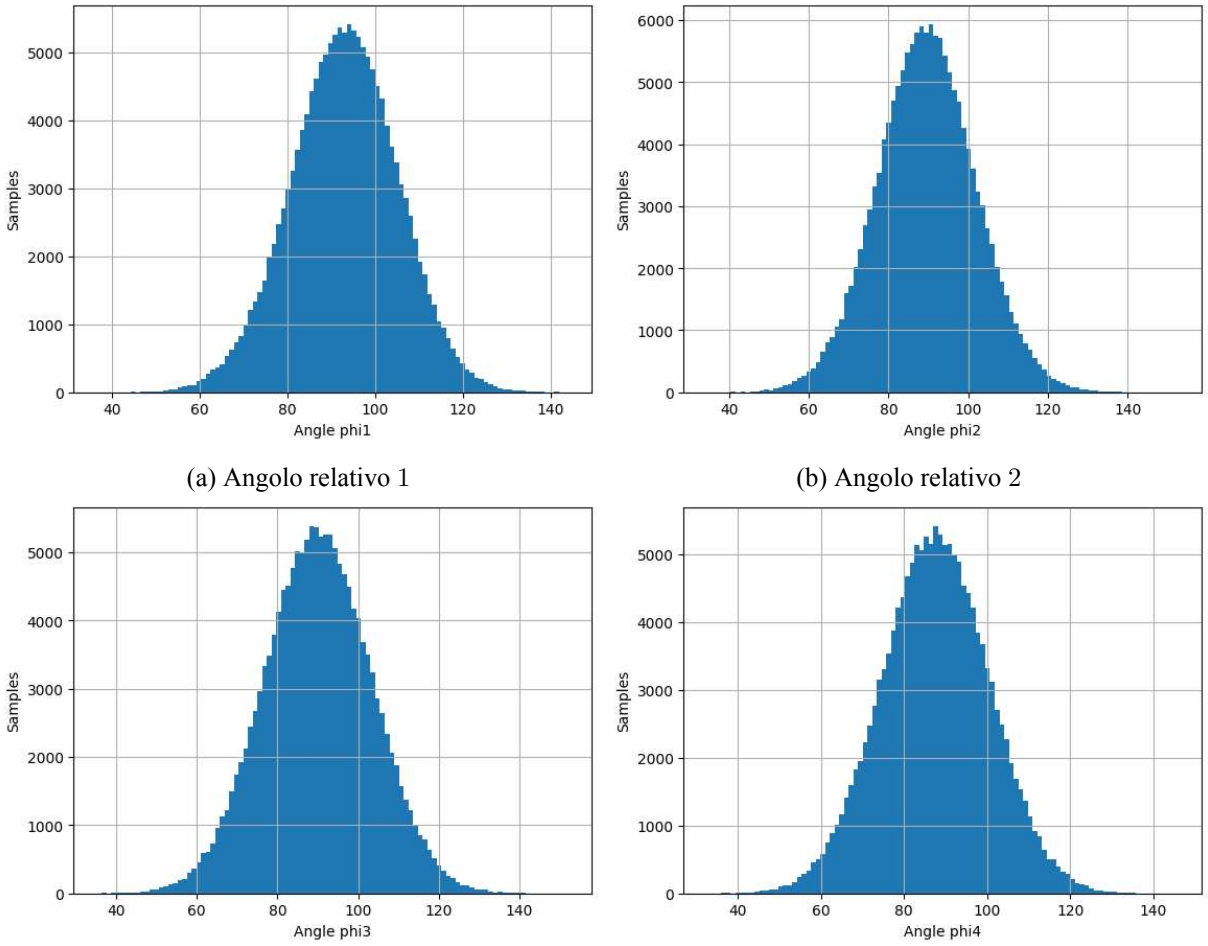
(a) Angolo relativo 1

(b) Angolo relativo 2

(c) Angolo relativo 3

(d) Angolo relativo 4

Figura 4.4: La distribuzione di tutti gli angoli relativi di ogni singola acquisizione, registrati durante la lunga misurazione effettuata dopo che l'algoritmo dv1 ha raggiunto livelli stabili, segue una distribuzione gaussiana centrata attorno all'angolo di 90°. Questo risultato indica una consistente stabilità nei valori acquisiti.



(a) Angolo relativo 1

(b) Angolo relativo 2

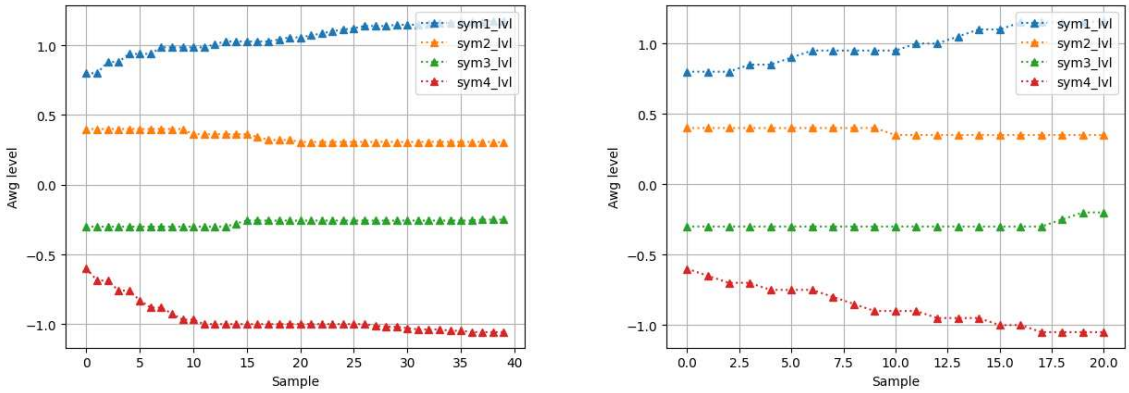
(c) Angolo relativo 3

(d) Angolo relativo 4

Figura 4.5: La distribuzione di tutti gli angoli relativi di ogni singola acquisizione, registrati durante la lunga misurazione effettuata dopo che l'algoritmo df1 ha raggiunto livelli stabili, segue una distribuzione gaussiana centrata attorno all'angolo di 90°. Questo risultato indica una consistente stabilità nei valori acquisiti.

4.2 Acquisizione con i livelli asimmetrici

In questa sezione saranno esaminate le prestazioni dell'algoritmo impiegando una configurazione di livelli asimmetrici. In particolare, verranno analizzati i risultati ottenuti con una condizione iniziale caratterizzata dai valori 0.8, 0.4, -0.3 e -0.6. L'analisi dei dati forniti contribuirà a verificare la capacità dell'algoritmo di adattarsi a configurazioni non simmetriche e di mantenere una buona performance nella configurazione della costellazione.



(a) L'andamento dei livelli della forma d'onda usando un passo variabile (dv)

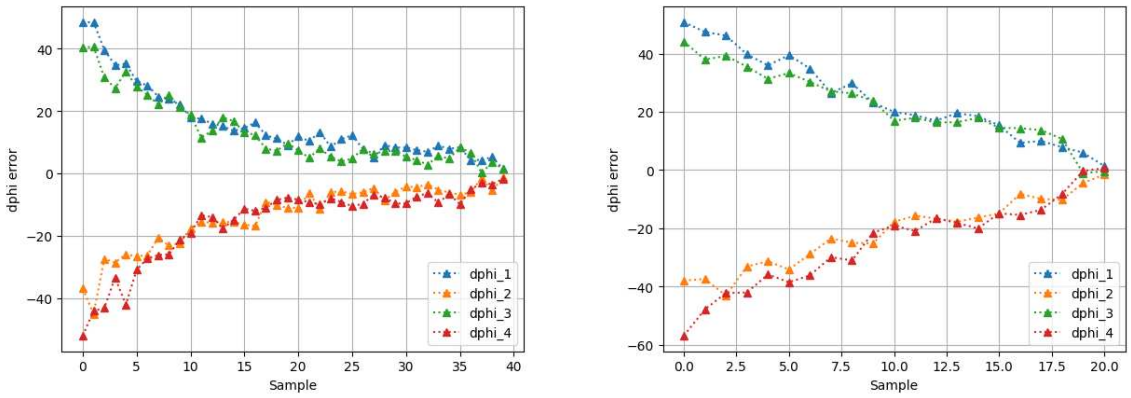
(b) L'andamento dei livelli della forma d'onda usando un passo fisso (df)

Figura 4.6: Modifiche ai livelli della forma d'onda apportate dall'algoritmo. Le variazioni illustrate mostrano l'impatto dell'algoritmo sui diversi livelli della forma d'onda, evidenziando il processo di generazione del segnale.

Nelle figure 4.6a e 4.6b sono presentati gli andamenti dei livelli settati dal DAC in funzione del numero delle iterazioni dell'algoritmo utilizzato. Nella seguente tabella 4.4 sono illustrati i livelli della forma d'onda raggiunti degli algoritmi nel test 2.

Livelli raggiunti da dv2	1.17	0.30	-0.25	-1.06
Livelli raggiunti da df2	1.16	0.35	-0.20	-1.05

Tabella 4.4: Livelli raggiunti dagli algoritmi nel test 2.

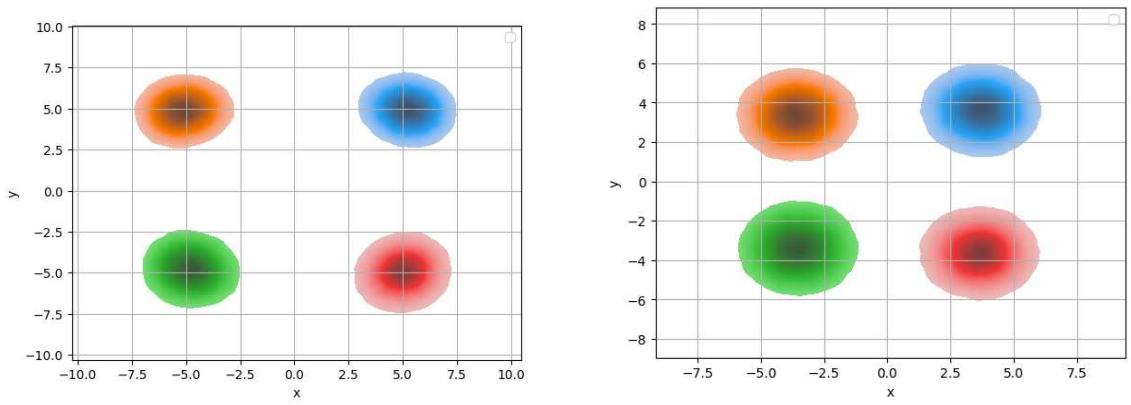


(a) L'andamento dell'errore con passo variabile nel test con i livelli asimmetrici (dv2)

(b) L'andamento dell'errore con passo fisso nel test con i livelli asimmetrici (df2)

Figura 4.7: Andamento dell'errore con gli algoritmi durante il test e si osserva come l'errore venga progressivamente minimizzato.

Nelle figure 4.7a e 4.7b sono presentate le convergenze degli errori rispetto agli angoli target degli algoritmi. In questo test, l'algoritmo con il passo variabile impiega 40 iterazioni per raggiungere una configurazione ottimale, raddoppiando il numero di iterazioni rispetto a quello con passo fisso.



(a) Costellazione della configurazione d'uscita (dv2)

(b) Costellazione della configurazione d'uscita (df2)

Figura 4.8: Costellazioni ottenute dalla media di 152.400 simboli, calcolata a partire dalla forma d'onda registrata alla condizione finale di uscita del rispettivo algoritmo. Questi dati illustrano la stabilità dei simboli generati, in quanto i cluster non sono sovrapposti.

Le costellazioni riportate nelle figure 4.8a e 4.8b sono ottenute sulla media di 152.400 simboli, a partire dalla forma d'onda risultante dalla condizione finale del rispettivo algoritmo. La media degli angoli relativi di tutti i simboli acquisiti durante questa lunga acquisizione è:

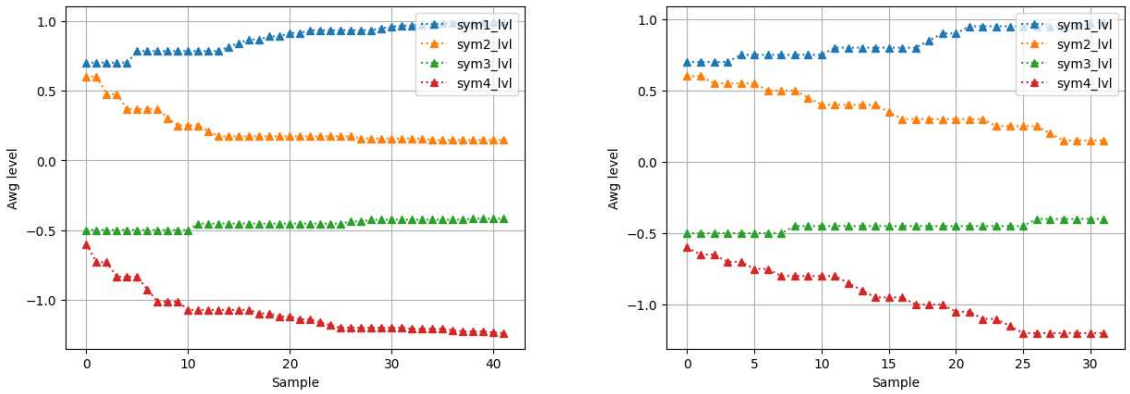
Media degli angoli relativi di dv2	93.34	88.46	90.54	87.65
Media degli angoli relativi di df2	92.35	87.40	90.87	89.37

Tabella 4.5: Media degli angoli relativi della lunga acquisizione della prova 2. La lunga acquisizione è composta da 300 misurazioni effettuate dal RFSoC, corrispondenti a un totale di 152.400 simboli acquisiti. Questi risultati mostrano come l'algoritmo riesca a raggiungere la configurazione ottimale.

Come riportato nella tabella 4.5, entrambi gli algoritmi sono stati in grado di ottimizzare la costellazione, con un errore massimo di $3.34°$ per il primo algoritmo e di $2.60°$ per il secondo.

4.3 Acquisizione con i livelli adiacenti e vicini alla saturazione

In questa sezione verranno analizzate le prestazioni dell'algoritmo utilizzando una configurazione di livelli adiacenti e prossimi al livello di saturazione. I test sono stati condotti con valori iniziali di 0.7, 0.6, -0.5 e -0.6.



(a) L'andamento dei livelli della forma d'onda usando un passo variabile (dv)

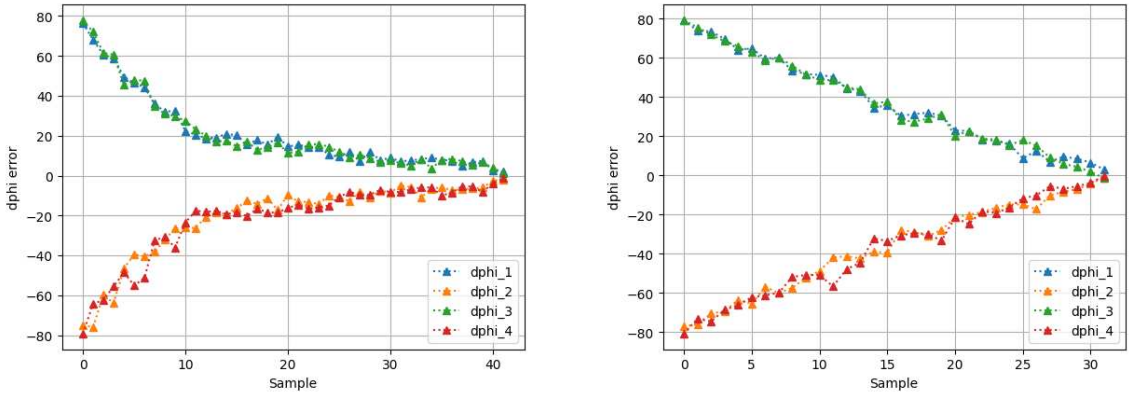
(b) L'andamento dei livelli della forma d'onda usando un passo fisso (df)

Figura 4.9: Modifiche ai livelli della forma d'onda apportate dall'algoritmo. Le variazioni illustrate mostrano l'impatto dell'algoritmo sui diversi livelli della forma d'onda, evidenziando il processo di generazione del segnale.

Nelle figure 4.9a e 4.9b sono presentati gli andamenti dei livelli settati dal DAC in funzione del numero delle iterazioni dell'algoritmo utilizzato. In questo test, i livelli non solo si avvicinano alla saturazione, ma sono anche molto prossimi tra loro. Come si può osservare nelle figure, l'algoritmo modifica progressivamente i rispettivi livelli per distanziarli fino a raggiungere le condizioni di uscita. Questo comportamento è diverso da quello osservato nei test precedenti, dove i livelli erano già sufficientemente distanziati. Nella seguente tabella 4.6 sono illustrati i livelli della forma d'onda raggiunti degli algoritmi nel test 3.

Livelli raggiunti da dv3	0.99	0.14	-0.42	-1.24
Livelli raggiunti da df3	0.98	0.15	-0.40	-1.20

Tabella 4.6: Livelli raggiunti dagli algoritmi nel test 3.

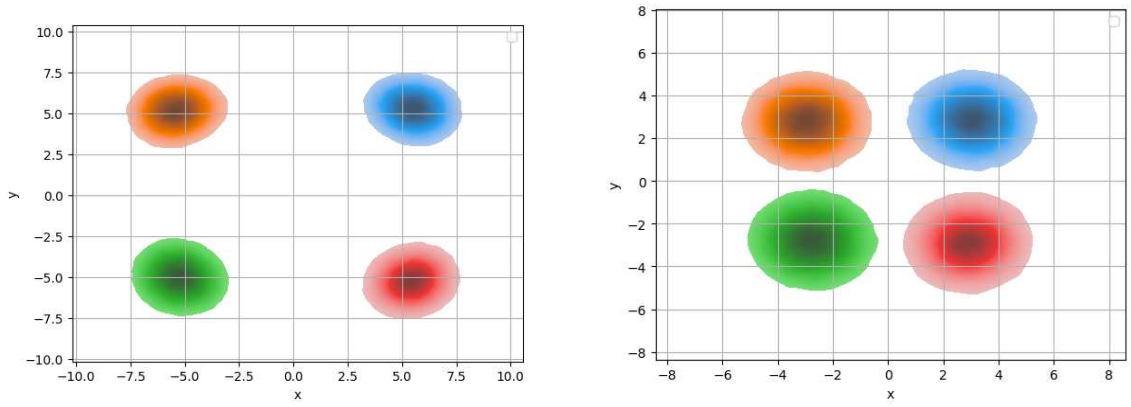


(a) L'andamento dell'errore con passo variabile nel test con i livelli adiacenti e vicini alla saturazione (dv3)

(b) L'andamento dell'errore con passo fisso nel test con i livelli adiacenti e vicini alla saturazione (df3)

Figura 4.10: Andamento dell'errore con gli algoritmi durante il test 3 e si osserva come l'errore venga progressivamente minimizzato.

Nelle figure 4.10a e 4.10b sono presentate le convergenze degli errori rispetto agli angoli target degli algoritmi. Inoltre, a causa della prossimità dei livelli della forma d'onda iniziale, si osserva un errore relativo maggiore rispetto agli altri test. Infatti, in questo caso, l'errore relativo parte da $80°$, mentre negli altri test iniziava da $40°$. In questo test, l'algoritmo con passo variabile richiede un numero maggiore di iterazioni rispetto all'algoritmo a passo fisso.



(a) Costellazione della configurazione d'uscita (dv3)

(b) Costellazione della configurazione d'uscita (df3)

Figura 4.11: Costellazioni ottenute dalla media di 152.400 simboli, calcolata a partire dalla forma d'onda registrata alla condizione finale di uscita del rispettivo algoritmo. Questi dati illustrano la stabilità dei simboli generati, in quanto i cluster non sono sovrapposti.

Le costellazioni mostrate nelle figure 4.11a e 4.11b sono ottenute dalla media di 152.400 simboli, calcolata partendo dalla forma d'onda risultante dalla condizione finale di ciascun algoritmo. La media degli angoli relativi di tutti i simboli acquisiti durante questa lunga acquisizione è:

Media degli angoli relativi di dv3	92.60	87.57	92.30	87.51
Media degli angoli relativi di df3	93.58	88.73	89.62	88.04

Tabella 4.7: Media degli angoli relativi della lunga acquisizione della prova 3. La lunga acquisizione è composta da 300 misurazioni effettuate dal RFSoC, corrispondenti a un totale di 152.400 simboli acquisiti. Questi risultati mostrano come l'algoritmo riesca a raggiungere la configurazione ottimale.

Come riportato dalla tabella 4.7, entrambi gli algoritmi hanno raggiunto una configurazione ottimale della costellazione, con un errore massimo di 2.60° per il primo algoritmo e di 3.58° per il secondo. I risultati di questo test dimostrano che entrambi gli algoritmi sono riusciti a ottimizzare la costellazione, nonostante la configurazione iniziale fosse particolarmente difficile a causa della vicinanza dei livelli tra loro e della soglia di saturazione.

4.4 Acquisizione con i livelli prossimi allo zero e un migliorato metodo di campionamento del segnale

In questa sezione verranno analizzate le prestazioni dell'algoritmo migliorato, applicato a livelli prossimi allo zero. I livelli considerati sono 0.2, 0, -0.1, e -0.4. Questo test ha l'obiettivo di valutare l'efficacia del nuovo metodo di campionamento nell'ottimizzazione dei livelli a bassa ampiezza.
L'algoritmo migliorato presenta un perfezionato metodo di campionamento. In particolare, si osservava uno scambio dei simboli dovuto all'inadeguatezza del punto di avvio del campionamento. Nella versione precedente del codice, il punto iniziale veniva determinato a partire dal primo massimo della correlazione tra la maschera e il segnale ideale. Tuttavia, a causa della vicinanza dei simboli e del rumore presente nel sistema, questo massimo non sempre corrisponde al massimo assoluto, provocando uno sfasamento dell'intero processo. Per ovviare a questo problema, la nuova versione del codice prevede la memorizzazione dei primi quattro massimi della correlazione e la selezione del massimo fra questi come punto iniziale, assicurando così la scelta del massimo assoluto. Da questo punto iniziale, viene calcolato il punto medio, e il processo torna a essere conforme a quello della versione precedente del codice.
Dopo aver risolto questa difficoltà, il nuovo codice ha dimostrato di essere capace di ottimizzare anche i livelli critici. I risultati ottenuti dall'algoritmo con il passo variabile.

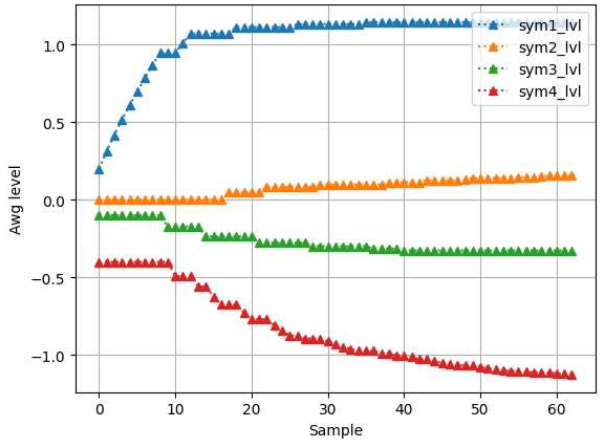


Figura 4.12: Modifiche ai livelli della forma d'onda apportate dall'algoritmo. Le variazioni illustrate mostrano l'impatto dell'algoritmo sui diversi livelli della forma d'onda, evidenziando il processo di generazione del segnale.

Nella figura 4.12 sono presentate le modifiche ai livelli della forma d'onda apportate dall'algoritmo in funzione del numero delle iterazioni nel test 4. In questo test, i livelli sono prossimi allo zero e anche molto prossimi tra loro. Come illustrato nella figura, l'algoritmo modifica progressivamente i rispettivi livelli per allontanarli tra loro e dallo zero, fino a raggiungere le

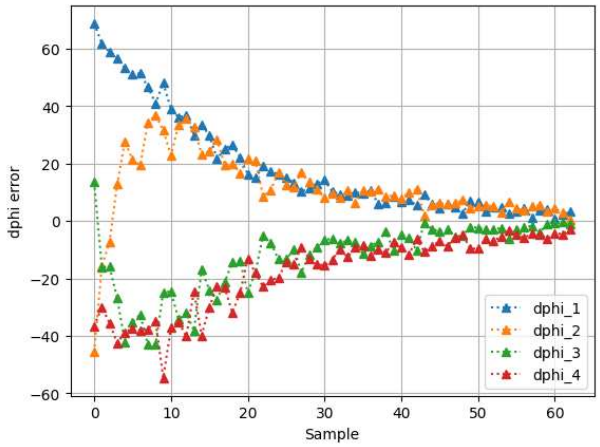


Figura 4.13: Andamento dell'errore con gli algoritmi durante il test 4 e si osserva come l'errore venga progressivamente minimizzato.

condizioni di uscita. Nella figura 4.13 è presentata la convergenza degli errori rispetto agli angoli target dell'algoritmo e si osserva che l'algoritmo impiega 63 iterazioni per raggiungere una configurazione ottima. Nella seguente tabella 4.8 sono illustrati i livelli della forma d'onda raggiunti degli algoritmi nel test 4.

Livelli raggiunti da dv4m	1.14	0.16	-0.33	-1.13

Tabella 4.8: Livelli raggiunti dagli algoritmi nel test 4.

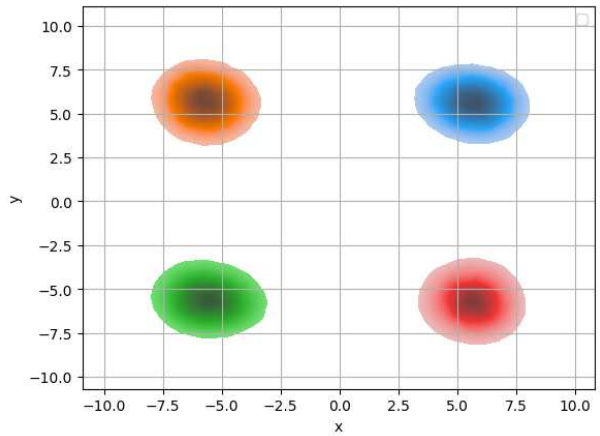


Figura 4.14: Costellazione della configurazione d'uscita in seguito a lunga acquisizione nel test 4. Questi risultati mostrano chiaramente che i cluster non sono sovrapposti.

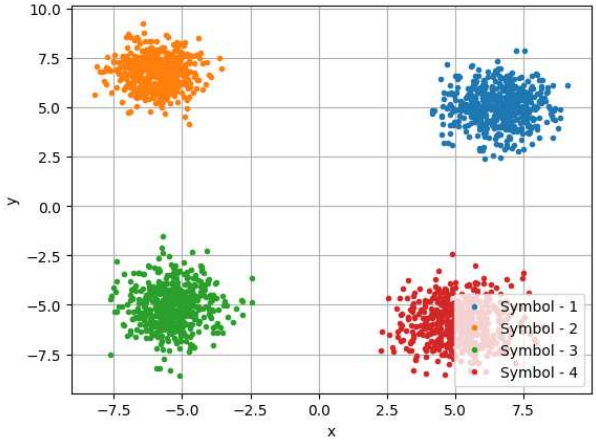


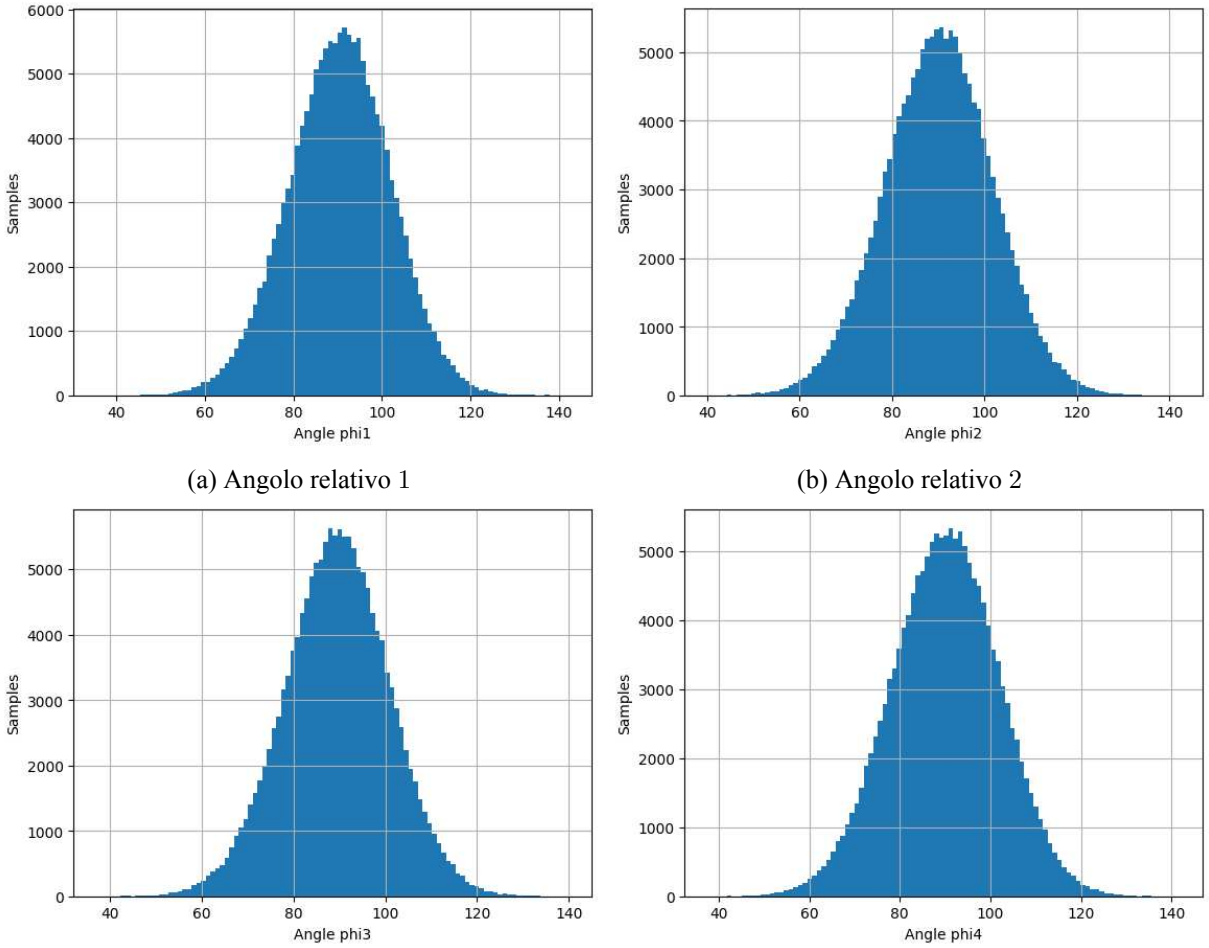
Figura 4.15: Costellazione ricavata dell'ultima acquisizione compiuta dall'algoritmo nel test 4. Questi risultati mostrano chiaramente come i cluster siano equamente distanziati.

La costellazione mostrata nella figura 4.14 è ottenuta sulla media di 152.400 simboli, partendo dalla forma d'onda trovata nella condizione d'uscita dell'algoritmo. Al contrario, la figura 4.15 rappresenta l'ultima costellazione fatta dall'algoritmo nell'ultima iterazione. In questo test l'algoritmo ha avuto successo, poiché i cluster della costellazione non sono sovrapposti e sono equi distanziati. Inoltre, l'errore massimo degli angoli relativi tra tutti i simboli ricevuti in questa lunga sessione di acquisizione è stato di 0.46°:

Media degli angoli relativi di dv4	90.46	90.13	89.45	89.96

Tabella 4.9: Media degli angoli relativi della lunga acquisizione della prova 4. La lunga acquisizione è composta da 300 misurazioni effettuate dal RFSoC, corrispondenti a un totale di 152.400 simboli acquisiti. Questi risultati mostrano come l'algoritmo riesca a raggiungere la configurazione ottimale.

Nella figura 4.16 è illustrato il grafico di tutti gli angoli relativi presi durante l'acquisizione lunga, che presenta un' andamento gaussiano, con il picco centrato sull'angolo di 90°. In questo test tutti i livelli sono molto ravvicinati, di conseguenza l'estrema vicinanza dei quattro livelli complica la possibilità di campionarli correttamente ed è stato necessario migliorare il metodo di campionamento per riuscire a ottimizzare in maniera adeguata la costellazione.



(a) Angolo relativo 1

(b) Angolo relativo 2

(c) Angolo relativo 3

(d) Angolo relativo 4

Figura 4.16: La distribuzione di tutti gli angoli relativi di ogni singola acquisizione, registrati durante la lunga misurazione effettuata dopo che l'algoritmo dv4m ha raggiunto livelli stabili, segue una distribuzione gaussiana centrata attorno all'angolo di 90°. Questo risultato indica una consistente stabilità nei valori acquisiti.

Capitolo 5

Conclusioni

L'obiettivo di questa tesi è stato l'ottimizzazione nella modulazione degli stati quantistici per un semi-device-independent continuous-variable quantum random number generator (SDI-CV-QRNG) basato su un sistema di rilevazione eterodina. In particolare, si è mirato a ottenere una costellazione di simboli generati tramite modulazione di fase, con simboli equidistanti di $90°$. Per raggiungere questo traguardo è stato sviluppato un algoritmo specifico in risposta alle imperfezioni del set-up sperimentale. L'algoritmo estrae le informazioni sugli angoli della costellazione dei simboli modulati e ricevuti e apporta modifiche a uno dei livelli della forma d'onda impiegata nella modulazione. L'algoritmo sviluppato ha dimostrato di essere efficace nell'ottimizzare tutte le forme d'onda proposte, riuscendo a portare gli angoli relativi a $90 \pm 4°$. Si è inoltre osservato che i quattro parametri `dphi` risultano accoppiati, con il primo accoppiato al terzo e il secondo al quarto. Di conseguenza, l'ottimizzazione di un parametro all'interno della coppia migliora automaticamente anche l'altro. Il grafico relativo all'errore mostra chiaramente come queste coppie di parametri siano speculari rispetto all'asse orizzontale.
In prospettiva futura, si potrebbe migliorare il metodo di campionamento dei dati ricevuti, con l'obiettivo di implementare una modulazione 8-PSK in sostituzione della modulazione Q-PSK attualmente utilizzata, con la possibilità di estendere successivamente a una modulazione 16-PSK. Tale sviluppo permetterebbe di realizzare QRNG sempre più veloci e competitivi, aumentando così le prestazioni dei SDI-QRNG rispetto ai modelli commerciali attualmente disponibili.

Bibliografia

[1] M. Herrero-Collantes e J. C. Garcia-Escartin, «Quantum random number generators,» *Reviews of Modern Physics*, vol. 89, 1 feb. 2017, issn: 15390756. doi: 10.1103/RevModPhys.89.015004.

[2] Y. Liu, Q. Zhao, M.-H. Li et al., «Device independent quantum random number generation,» lug. 2018. doi: 10.1038/s41586-018-0559-3. indirizzo: http://arxiv.org/abs/1807.09611%20http://dx.doi.org/10.1038/s41586-018-0559-3.

[3] V. Mannalath, S. Mishra e A. Pathak, «A Comprehensive Review of Quantum Random Number Generators: Concepts, Classification and the Origin of Randomness,» mar. 2022. doi: 10.1007/s11128-023-04175-y. indirizzo: http://arxiv.org/abs/2203.00261%20http://dx.doi.org/10.1007/s11128-023-04175-y.

[4] X. Ma, X. Yuan, Z. Cao, B. Qi e Z. Zhang, *Quantum random number generation*, 2016. doi: 10.1038/npjqi.2016.21.

[5] Y. Li, Y. Fei, W. Wang et al., «Practical security analysis of a continuous-variable source-independent quantum random number generator based on heterodyne detection,» *Optics Express*, vol. 31, p. 23 813, 15 lug. 2023, issn: 10944087. doi: 10.1364/oe.493586.

[6] M. Avesani, H. Tebyanian, P. Villoresi e G. Vallone, «Semi-Device-Independent Heterodyne-Based Quantum Random-Number Generator,» *Physical Review Applied*, vol. 15, 3 mar. 2021, issn: 23317019. doi: 10.1103/PhysRevApplied.15.034034.

[7] B. E. A. Saleh, *Fundamentals of photonics*. Wiley, 2019, p. 1370, isbn: 9781119506874.

[8] P. Y. Amnon Yariv, *Photonics: Optical Electronics in Modern Communications*. Oxford University Press, Inc.198 Madison Ave. New York, NYUnited States, 2006, isbn: 9780195179460.

[9] N. Benvenuto e M. Zorzi, *Principles of communications networks and systems*. Wiley, 2011, p. 786, isbn: 9780470744314.

[10] U. Leonhardt, *Measuring the Quantum State of Light*. Cambridge: Cambridge University Press, 1997, isbn: 9780521497305.

[11] D. J. Griffiths e D. F. Schroeter, *Introduction to quantum mechanics*, Third edition. Cambridge ; New York, NY: Cambridge University Press, 2018, isbn: 978-1-107-18963-8.

[12] C. Gerry e P. Knight, *Introductory Quantum Optics*. London: Cambridge University Press, 2004, isbn: 9780511791239. doi: https://doi.org/10.1017/CBO9780511791239.

[13] J. B. Brask, A. Martin, W. Esposito et al., «Megahertz-Rate Semi-Device-Independent Quantum Random Number Generators Based on Unambiguous State Discrimination,» *Physical Review Applied*, vol. 7, 5 mag. 2017, issn: 23317019. doi: 10.1103/PhysRevApplied.7.054018.

[14] H. Tebyanian, M. Avesani, G. Vallone e P. Villoresi, «Semi-device-independent randomness from d -outcome continuous-variable detection,» *Physical Review A*, vol. 104, 6 dic. 2021, issn: 24699934. doi: 10.1103/PhysRevA.104.062424.