

Capitolo 1:

Qubit e IBM Quantum Experience

In questo primo capitolo sarà fornita un'approfondita descrizione di cosa sia un qubit e verranno descritte le possibilità fornite dall'IBM con il progetto di *IBM Quantum Experience*, in modo da avere tutti gli strumenti necessari per comprendere cosa sia stato svolto in questo lavoro di tesi.

1.1 Qubit

Come il *bit* è l'unità fondamentale della computazione classica, così il *quantum bit* (o semplicemente *qubit*) è l'unità fondamentale dell'informazione e della computazione quantistica [1]. Quest'analogia sarà la linea guida seguita per descrivere cosa sia un qubit. Il primo approccio per la descrizione di un qubit è puramente matematico, ignorando, per il momento, come questo possa essere costruito fisicamente. Si discute, dunque, un oggetto matematico.

L'oggetto in questione è quello che in meccanica quantistica viene chiamato *sistema a due livelli*, ossia una generica sovrapposizione (o combinazione lineare) di due stati normalizzata [2]. Usando la notazione di Dirac e indicando i due vettori (gli stati di cui prima) della base di uno spazio di Hilbert di dimensione 2 con $|0\rangle$ e $|1\rangle$, qualsiasi combinazione di questi ammissibile è scrivibile come:

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

dove ψ è lo stato in cui il sistema si trova, α e β coefficienti complessi che rispettano il vincolo di normalizzazione:

$$|\alpha|^2 + |\beta|^2 = 1$$

Per questo motivo ci si riferisce spesso a questi coefficienti come *ampiezze di probabilità*, in quanto, misurando un'osservabile A avente come autostati (o autovettori) $|0\rangle$ e $|1\rangle$ il loro modulo quadro indica proprio la probabilità di misurare l'uno o l'altro autostato.

Inoltre, dato che $|0\rangle$ e $|1\rangle$ costituiscono la base dello spazio considerato, questi autovettori godono di *ortonormalità*:

$$\langle 0|0\rangle = 1 \quad \langle 1|1\rangle = 1$$

$$\langle 0|1\rangle = 0 \quad \langle 1|0\rangle = 0$$

Per comodità successiva e semplicità di interpretazione, si introduce ora una rappresentazione più geometrica di un sistema a due livelli: la *sfera di Bloch*. La possibilità di usare un'interpretazione così geometrica è figlia diretta della condizione di normalizzazione. Si può infatti scrivere il generico stato $|\psi\rangle$ come:

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \right)$$

la quale, tenendo conto che un fattore di fase *overall* come è $e^{i\gamma}$ non può influenzare alcun modo le osservabili fisiche, si può semplificare come segue:

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

dove $0 \leq \theta < \pi$ e $0 \leq \varphi < 2\pi$. Proprio questi due parametri reali definiscono un punto nella rappresentazione grafica (fig 1.1) della sfera di Bloch, se si considera θ che il vettore-stato forma con l'asse delle Z e φ l'angolo azimutale.

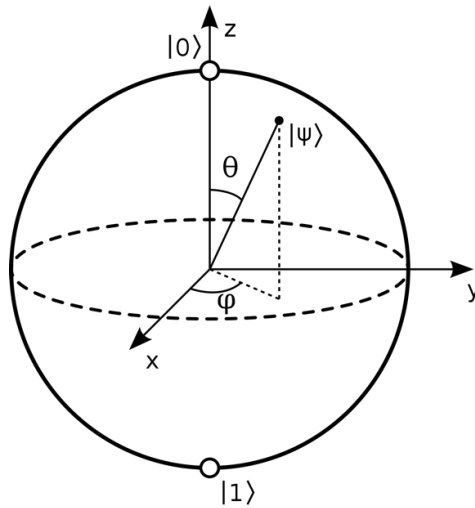


Fig 1.1: Rappresentazione della sfera di Bloch

È ora il momento di discutere di quanta informazione può essere contenuta in un qubit. Il quesito non è banale; è anzi cruciale per capire se il quantum computing possa davvero offrire vantaggi rispetto all'informatica classica. Il fatto è che, sebbene ci sia un'infinità di punti sulla superficie della sfera unitaria, il risultato di un'osservazione può restituire solamente $|0\rangle$ o $|1\rangle$ con le probabilità discusse prima. Non solo, l'operazione di misura è proiettiva, quindi la sovrapposizione di stati viene istantaneamente persa quando questa è applicata! In meccanica quantistica si parla di “collasso della funzione d'onda”, un fenomeno non ancora spiegato, ma postulato, e irreversibile. Sarà ora evidente anche al lettore meno afferrato in meccanica quantistica perché la stesura di algoritmi quantistici necessitino di una logica completamente diversa da quella dell'informazione classica.

Per completezza, ma argomento che esula dalle intenzioni di questo lavoro di tesi, si vuole dire che sì, il quantum computing ha grossi vantaggi rispetto alle tecniche tradizionali in termini di *costi computazionali*; anche in virtù del fatto che sfrutta altri fenomeni della meccanica quantistica che emergono in presenza di più sistemi (qubit): *entanglement* e interferenza.

1.2 Operazioni su singolo qubit

Si continua in questo paragrafo un parallelo tra informazione classica e quantistica per esaltare le maggiori possibilità che questa seconda comporta.

Nell'informatica tradizionale, l'unica porta logica non banale che si può applicare ad un singolo bit è la porta NOT che opera nel modo che segue:

$$\begin{aligned}0 &\rightarrow 1 \\1 &\rightarrow 0\end{aligned}$$

Il compito che assolve questa porta è sostanzialmente quello di scambiare 0 con 1, e viceversa. Non ci sono altre operazioni non banali (e.g. $1 \rightarrow 1$) che si possono fare su un singolo bit. Il motivo è semplice: si hanno a disposizione due soli stati.

E nel quantum computing? Nel quantum computing l'infinità di stati che un qubit può assumere comporta altrettante operazioni applicabili ad esso.

Chiaramente esistono delle analoghe porte logiche, chiamati più comunemente *gate* in questo contesto. Si hanno infatti l'operazione d'identità

$$\begin{aligned}|0\rangle &\rightarrow |0\rangle \\|1\rangle &\rightarrow |1\rangle\end{aligned}$$

e l'analogo NOT gate

$$\begin{aligned}|0\rangle &\rightarrow |1\rangle \\|1\rangle &\rightarrow |0\rangle\end{aligned}$$

È necessario qui far notare che questi gate sono in realtà molto più generali. Infatti, questi possono agire su qualsiasi stato in cui si trova il qubit, dunque l'identità è scrivibile come:

$$|\psi\rangle \rightarrow |\psi\rangle$$

Per apprezzare come agisce il NOT gate, chiamato anche X gate per ragioni che seguono, è conveniente introdurre la notazione matriciale secondo la quale

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In questa notazione l'X gate è rappresentato dalla matrice $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Per cui, nel caso di un generico stato $\alpha|0\rangle + \beta|1\rangle$, esso agisce così:

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

Ora, prima di mostrare altri esempi di gate, si vuole rispondere ad una domanda che qui sorge spontanea: per accedere ad uno qualsiasi degli infiniti punti/stati della sfera di Bloch, quanti gate servono? Infiniti? Fortunatamente, il numero di gate necessari è finito. Si può provare l'esistenza di *set universale di gate*, ossia un numero finito di gate che permettono di approssimare con precisione

asintotica qualsiasi operazione si intenda applicare sullo stato. In altri termini: dato un punto sulla sfera di Bloch, si può raggiungere qualsiasi altro punto con una combinazione finita dei gate appartenenti al set universale. Più in generale, nel caso di sistemi a più qubit, si vuole saper implementare “qualsiasi trasformazione unitaria” il che include anche l’*entanglement*, ma questo non riguarda gli esperimenti a singolo qubit.

In seguito, una lista di gate comunemente usati nel quantum computing:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{2}} \end{bmatrix}$$

Tra questi, Y e Z inducono una rotazione di π radianti attorno al rispettivo asse; H è noto in letteratura come *Hadamard gate*, I è l’identità (non apporta alcuna modifica allo stato del qubit), P è il *phase gate*, dunque altera la fase relativa tra $|0\rangle$ e $|1\rangle$ dello stato attuale e S non è che un caso particolare del P gate.

Tralasciando momentaneamente il fatto che l’esistenza di un set universale è fondamentale per il quantum computing, in quanto il più delle volte si è in grado implementare fisicamente solo certi gate, idealmente, qualsiasi operazione unitaria su singolo qubit può essere rappresentata da un gate a tre parametri: l’U3 gate; la cui forma matriciale risulta essere:

$$U3(\theta, \varphi, \lambda) = \begin{pmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\varphi} \sin \frac{\theta}{2} & e^{i(\varphi+\lambda)} \cos \frac{\theta}{2} \end{pmatrix}$$

1.3 Realizzare un qubit

In questo paragrafo si illustra come vengono realizzati fisicamente i qubit e come vengono controllati; un passaggio dal teorico al pratico che storicamente ha richiesto decine di anni (Feynman propose il concetto di *quantum computer* già nel 1981!). Non verranno descritte le tecniche di costruzione di un qubit, in quanto non si intende approfondire l’aspetto ingegneristico di questo arduo compito, si illustreranno “semplicemente” dei principi guida.

La stesura dei criteri che deve soddisfare un sistema fisico per poter essere considerato l’*hardware* di un computer quantistico la si deve a Davide DiVincenzo [3]; il quale, nel 2000, propose queste cinque condizioni necessarie:

Possibilità di identificare qubit ben definiti ed aumentarli di numero. Questo criterio richiede una solida rappresentazione dell’informazione.

Capacità di inizializzare lo stato del sistema: portare con estrema efficacia i qubit in un determinato stato iniziale e, solo dopo, eseguire un calcolo.

Tempi di decoerenza sufficientemente lunghi.

Esistenza di una classe universale di porte logiche quantistiche (gate) per il controllo preciso dei qubit; sostanzialmente, essere in grado di implementare tutti i gate appartenenti ad un set universale.

Effettuare una misurazione per ottenere il risultato del calcolo eseguito e trasmettere l’informazione quantistica.

Ora un breve e doveroso commento a questi criteri.

“Possibilità di identificare qubit ben definiti ed aumentarli di numero”: riprendendo quanto discusso nei paragrafi precedenti, un qubit è un sistema a due livelli ed ogni sistema fisico scelto per realizzarlo deve assicurarci il fatto che questo non esca mai dal sottospazio (di Hilbert) di questi due livelli. Per quanto già questo sia di difficile costruzione, “aumentarli di numero” è la parte di questo criterio che attualmente rappresenta la sfida più ardua da superare.

“Capacità di inizializzare lo stato del sistema”: questa richiesta, piuttosto semplice nell’informazione classica (è equivalente a richiedere di essere in grado di settare un bit a 0), non è banale nel mondo quantistico. Si noti che la richiesta è quella di saper inizializzare *uno* stato iniziale degli infiniti che costituiscono la sfera di Bloch. Dal punto di vista teorico non esiste alcuno stato privilegiato, dal punto di vista pratico, generalmente, si preferisce inizializzare un qubit allo stato fondamentale $|0\rangle$. D’altro canto, gli stessi assiomi della meccanica quantistica vengono in soccorso in questo: dato che la misura è proiettiva, ogni volta che misuriamo lo stato di un qubit sulla base $[|0\rangle; |1\rangle]$ e otteniamo $|0\rangle$ si è certi che immediatamente dopo la misura lo stato fisico del sistema sia $|0\rangle$. Attenzione: saper effettuare una misura è la richiesta del quinto criterio.

“Tempi di decoerenza sufficientemente lunghi”: questa richiesta, non sempre incluso tra i criteri di DiVincenzo nei libri testo, rispecchia un *leitmotiv* della fisica: la realtà presenta dei “difetti” rispetto al mondo ideale. I difetti che non possiamo trascurare, in questo caso, sono la non stazionarietà del livello eccitato $|1\rangle$ (tende a decadere allo stato di *ground* $|0\rangle$) e l’interazione del sistema con l’ambiente. I tempi di decoerenza di un qubit saranno oggetto di discussione del capitolo 2 come parametro fondamentale per la caratterizzazione; l’espressione “sufficientemente lunghi” va intesa in relazione alla durata dei gate applicati al qubit e al numero di operazioni necessarie per svolgere un determinato algoritmo.

“Esistenza di una classe universale di porte logiche quantistiche”: sebbene l’esistenza di un set universale di gate sia assicurato da un teorema, l’implementazione dei gate appartenenti a questo non va considerata un *task* semplice da affrontare anche in virtù del fatto che gli algoritmi di quantum computing richiedono spesso un’elevata precisione dei gate.

“Effettuare una misurazione”: si richiede di essere in grado di trasformare un’informazione quantistica in informazione classica. Non solo, bisogna essere in grado di farlo con un’adeguata accuratezza.

I criteri di DiVincenzo non citano alcuna tecnica di costruzione e nemmeno una modalità di controllo di un qubit. La mancanza di questi vincoli permette di teorizzare qualsiasi algoritmo quantistico senza che effettivamente esista un computer quantistico, di fatto alcuni algoritmi precedono la prima realizzazione di un computer di questo genere, e inoltre lascia libertà nella scelta di sistemi fisici adatti a questi scopi.

Negli anni, son stati creati qubit con fotoni, “ioni intrappolati” (preferibile l’espressione inglese *ion in traps*), circuiti superconduttori, difetti nei reticoli di diamante (NV diamonds), quantum dots, risonanza magnetica nucleare in molecole (*NMR in molecules*) ecc.

Di questi si potrebbe discutere diffusamente di pro e contro, ma ai fini di questa tesi si intende fornire una semplice e schematica (fig1.2) classificazione in termini di tempi di coerenza e velocità prima di trattare singolarmente i qubit realizzati con circuiti superconduttori; non perché si vuole affermare che siano i migliori, ma semplicemente perché sono quelli usati da IBM e, di conseguenza, quelli usati nella parte sperimentale di questo lavoro.



Fig 1.2: Classificazione presa da [4]. In questo caso, “velocità” indica la durata di una singola operazione (gate).

Attualmente, i leader mondiali nella costruzione di computer quantistici basati su architettura di circuiti superconduttori sono Google e IBM. Google, con *Sycamore*, vanta la costruzione di un computer a 53 qubit [5]. Con l’impiego di questo backend, l’azienda statunitense ha annunciato di aver raggiunto la *quantum supremacy* nell’ottobre del 2019 [6]. Un numero maggiore di qubit è stato raggiunto da IBM con due diversi backend, *ibmq-manhattan* e *ibmq-brooklyn*, entrambi computer da 65 qubit e *quantum volume* pari a 32. Questo parametro, il quantum volume, è una possibile metrica per misurare le prestazioni di un computer quantistico. È un numero intero stimato in base alla complessità dei circuiti che un computer può effettivamente implementare. Tenzialmente, il suo valore cresce col numero dei qubit, sebbene in realtà il suo calcolo si basa sui valori di fedeltà delle operazioni, sul grado di connettività tra i diversi qubit, sul set di gate accuratamente calibrati e sull’efficacia di mappatura del circuito [7]. Di fatti, IBM dichiara di aver raggiunto valori di quantum volume maggiori con computer a 27 qubit, quali *ibmq-montreal* e *ibmq-dublin*, per i quali è stato stimato essere pari a 128 e 64, rispettivamente. Invece, il backend utilizzato per questa tesi, *ibmq-armonk*, è a singolo qubit e ha quantum volume pari a 1.

Circa il confronto tra computer quantistici che si basano su tecnologie diverse, si riportano valori tipici di velocità e tempi di decoerenza. Quelli basati su ion traps hanno valori tipici di durata di singolo gate tra i $0,1\mu s$ e i $5\mu s$ con valori che crescono per operazioni su più qubit ($50-3000\mu s$) [8]; per questo tipo di qubit si è raggiunto un tempo di coerenza di circa 6000s, oltre un’ora [9]. Per i qubit superconduttori, invece, si attestano valori nettamente inferiori sia per il tempo di implementazione di un gate, sia per il tempo di coerenza. Rispettivamente, valori tipici possono essere di decine di nanosecondi e centinaia di microsecondi; che comunque consentono di poter applicare un numero di operazioni nell’ordine di $10^3 - 10^4$; quelle richieste dai criteri di DiVicenzo.

Uno dei motivi per cui multinazionali, come IBM e Google per l'appunto, reputino i computer quantistici a superconduttori preferibili è che questi promettono maggiori possibilità in termini di scalabilità. Sono infatti gli unici ad aver raggiunto le decine di qubit e, al tempo stesso, avere un'architettura che permetta l'implementazione di sistemi ancora più complessi. Caratteristica estremamente interessante, dato che la potenza di calcolo di questi computer è esponenziale in funzione del numero di qubit. Dal blog [10] di IBM si legge che sono in corso i lavori per la costruzione di un device da 1121 qubit, *ibmq-condor*, il termine dei quali è previsto nel 2023.

1.4 Qubit superconduttori

Come annunciato, si approfondisce qui il caso di qubit realizzati con circuiti superconduttori [11]. L'idea è quella di realizzare un oscillatore armonico con un circuito LC, dove compaiono due elementi di circuito: l'induttore (di induttanza L) e il capacitore (di capacità C). Come noto, le energie contenute in questi due elementi sono:

$$E_C = \frac{1}{2} CV^2 = \frac{Q^2}{2C} \quad E_L = \frac{1}{2} LI^2 = \frac{\phi^2}{2L}$$

dove $Q=CV$ è la carica del capacitore e $\phi=LI$ è il flusso di campo magnetico nell'induttore. Segue che l'Hamiltoniana classica è la somma di questi due termini:

$$H_{Cl} = \frac{Q^2}{2C} + \frac{\phi^2}{2L} = \frac{C^2}{2C} + \frac{1}{2} C \omega_0^2 \phi^2$$

in particolare, nella seconda formulazione è stato introdotto il parametro $\omega_0 = 1/\sqrt{LC}$ per potere identificare più facilmente un'Hamiltoniana di "massa" C , momento Q , "posizione" ϕ e frequenza ω_0 . In questo modo si può riscrivere H_{Cl} in termini di operatori di *creazione* (a) e *distruzione* (a^\dagger) come noto dalla meccanica quantistica. Sostituiamo quindi Q e ϕ coi rispettivi operatori in termini di a e a^\dagger .

$$Q \rightarrow \hat{Q} = i \sqrt{\frac{\hbar}{2Z_0}} (a - a^\dagger) \quad \phi \rightarrow \hat{\phi} = i \sqrt{\frac{\hbar Z_0}{2}} (a + a^\dagger)$$

dove $Z_0 = \sqrt{L/C}$ è l'impedenza. Si ricorda che valgono le regole di commutazione:

$$[a, a^\dagger] = 1 \quad [\hat{Q}, \hat{\phi}] = i \hbar$$

Si è ora in grado di riscrivere:

$$\widehat{H}_{LC} = \hbar \omega_0 (a a^\dagger + \frac{1}{2}) \quad \widehat{H}_{LC} |n\rangle = E_n |n\rangle$$

ora $|n\rangle$ sono gli autostati e $E_n = \hbar \omega_0 (n + 1/2)$ gli autovalori dell'Hamiltoniana scritta. Siccome la differenza di energia di un livello energetico e il successivo è sempre pari a $\hbar \omega_0$, è impossibile

selezionare solo due stati (condizione necessaria per il qubit). Si sceglie, dunque, di introdurre dei termini non-lineari così da rompere la regolarità dei “salti” di energia e la ΔE_{1-2} è unica.

Per ottenere questo risultato, il circuito è reso anarmonico sostituendo l'induttore L con una giunzione Josephson. Una giunzione Josephson consiste di due superconduttori separati da un isolante. Il suo funzionamento si basa poi sull'effetto tunnel della coppia di Cooper attraverso lo strato isolante.

Rimandando all'Appendice la trattazione dell'Hamiltoniana di questo sistema e i calcoli necessari per ottenere i nuovi livelli energetici (usando la teoria delle perturbazioni) si illustrano in fig 1.3 gli schemi delle giunzioni Josephson e i grafici dei livelli energetici del sistema armonico (non perturbato) e del sistema anarmonico (perturbato).

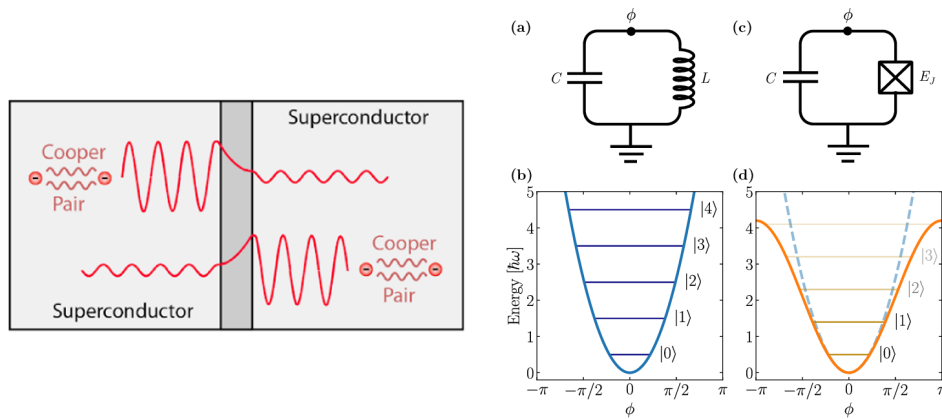


Fig 1.3: A destra un'idea di funzionamento della giunzione Josephson in cui si può apprezzare una rappresentazione qualitativa dell'effetto tunnel. A sinistra gli schemi di un circuito LC (oscillatore armonico) e di un circuito con SQUID in serie ad un capacitore e i rispettivi livelli energetici.

1.5 IBM Quantum Experience

IBM Quantum Experience è un servizio cloud messo a disposizione da IBM dal quale è possibile accedere a computer quantistici. Con un account “base” e gratuito è possibile utilizzare hardware composti di 1 o 5 qubit oppure sfruttare dei simulatori di sistemi formati da 32 fino a 5000 qubit; ai primi non si ha accesso diretto e immediato: bisogna inviare al cloud l'algoritmo che si intende eseguire, la richiesta viene messa in coda alle altre di altri utenti e solo successivamente viene eseguita; per i simulatori, invece, non c'è alcuna lista d'attesa [12]. I qubit usati da IBM sono superconduttori di tipo transmon, con i quali si può interagire grazie a due *framework* open source: Qiskit e Qiskit Pulse; entrambi in linguaggio Python.

Con Qiskit si può creare qualsiasi circuito quantistico usando un'interfaccia grafica (*composer*) o direttamente programmando in Python. Il composer è dotato di cinque righe, rappresentati 5 qubit diversi inizializzati allo stato fondamentale $|0\rangle$ ai quali si possono applicare diversi gate scegliendo da una banda laterale. Da questa banda si può importare anche l'operazione di misura del qubit per salvare l'informazione quantistica su un bit classico e ottenere così dell'informazione classica utilizzabile. Solitamente, il composer è utilizzato più per scopo didattico in quanto non offre tutte le possibilità che si hanno programmando con le librerie di Qiskit. Dico “per scopo didattico” anche perché l'interfaccia grafica mostra come si modifica lo stato del qubit sulla sfera di Bloch ogni qualvolta si aggiunge un gate e dà una stima percentuale delle misure di $|0\rangle$ o $|1\rangle$ che si otterrebbero facendo girare il circuito creato su un computer quantistico. Questo è sicuramente un comodo

approccio al mondo del quantum computing per chi non è esperto di meccanica quantistica o non ha affinità con gli operatori. Inoltre, grazie al fatto che si possono far girare i programmi-circuiti realizzati sia su un simulatore che su un vero computer quantistico, ci si può far un'idea della discrepanza che c'è tra “computer ideale” (rappresentato dal simulatore, il quale assume la totale assenza di errori e rumore elettronico) e “computer reale” osservando i risultati ottenuti dall'hardware quantistico.

Qiskit Pulse, invece, permette di lavorare con i quantum computer ad un livello più vicino all'hardware. Non va dimenticato che per implementare fisicamente un circuito quantistico stilizzando dei qubit superconduttori richiede saper “tradurre” delle istruzioni informatiche (esempio banale: applicare un X gate) in impulsi di microonde. Qiskit Pulse permette proprio questo: conoscere la corrispondenza gate-impulsi. Nel capitolo successivo, che riguarda la caratterizzazione di un qubit, verrà descritto un programma che usa solamente le librerie di Qiskit Pulse e si mostrerà come implementare sia l'X gate, ottimizzando un impulso gaussiano per passare da $|0\rangle$ a $|1\rangle$ e viceversa, sia l'Hadamard gate; col quale, si ricorda si passa dallo stato $|0\rangle$ alla combinazione lineare $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$.

Vale la pena qui introdurre il *backend* utilizzato in questo lavoro di tesi: `ibmq_armonk`. È il più “piccolo” dei computer messi a disposizione da IBM [13], consta di un solo qubit e deve il suo nome alla città in cui ha sede IBM. Le caratteristiche principali di questo computer sono:

- Tipo di processore: Canary r1.2
- Base di gate: I, RZ, SX, X
- Errore di readout medio: 2.630e-2
- T1 medio: 211.96 μs
- T2 medio: 223.56 μs
- Frequenza di risonanza: 4.972 GHz

Il lavoro di calibrazione del qubit viene svolto giornalmente da IBM, aggiornando così i valori appena mostrati i quali possono piccole variazioni dovute ad alterazioni delle condizioni dell'ambiente di lavoro (ad esempio un aumento della temperatura interna al criostato in cui sono posti i qubit). Una calibrazione frequente permette di avere dei gate sempre ottimizzati alle condizioni attuali del qubit e, dunque, ad avere le migliori prestazioni ottenibili dallo stesso. Un modo per poter svolgere questo lavoro automaticamente con Qiskit Pulse è mostrato nel capitolo che segue.

Capitolo 2:

Caratterizzazione di un qubit

L'esperimento di caratterizzazione di un qubit qui proposto prevede l'utilizzo di un programma scritto in linguaggio Python. Tale programma utilizza le librerie di Qiskit [14] e QiskitPulse [15] per controllare un hardware quantistico messo a disposizione dall'IBM sulla piattaforma IBM-quantum-lab; in particolare, è stato utilizzato il backend *ibmq_armonk*.

Senza entrare nel dettaglio di ogni singola funzione implementata, viene di seguito descritta la procedura di caratterizzazione; il programma completo è riportato in appendice. Questo esperimento assume solo la conoscenza di una precedente stima di frequenza di risonanza; nel caso non la si conoscesse, cambierebbe solo la prima parte di esperimento: gli impulsi mandati in ingresso al qubit dovrebbero ricoprire un grande intervallo di frequenze entro il quale è ragionevole pensare che si trovi la risonanza, questo intervallo potrebbe essere scelto in base alle caratteristiche di costruzione del qubit stesso. Valori tipici si attestano sui 4-5GHz.

Ogni valore numerico che compare in questo capitolo non ha alcuna validità generale; in quanto, queste procedure di calibrazione devono essere svolte quotidianamente su ogni qubit di qualsiasi device quantistico.

Frequenza di risonanza

Il primo step necessario per la calibrazione è trovare la frequenza di risonanza del qubit e dunque la differenza di energia tra lo stato fondamentale $|0\rangle$ e il primo stato eccitato $|1\rangle$; gli unici due stati che si assumono accessibili in questo tipo di esperimento. Per farlo, si inviano impulsi di microonde in un range di frequenze centrato alla frequenza precedentemente stimata. Per ogni impulso in ingresso, si misura il segnale in uscita da qubit aspettandosi di osservare un picco: è proprio quest'osservazione che ci permette di stimare la frequenza di risonanza. La fig 2.1 mostra l'andamento del segnale letto in uscita in funzione delle frequenze degli impulsi inviati.

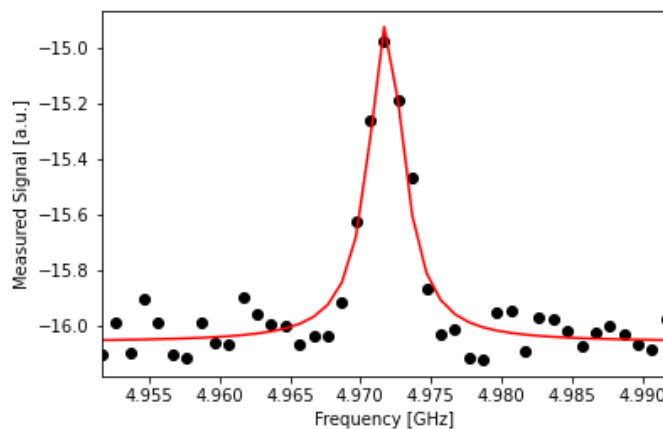


Fig 2.1: il grafico mostra un picco molto pronunciato giusto al centro del range di frequenze inviate in ingresso, l'ascissa di questo picco rappresenta la prima stima di frequenza di risonanza.

I dati sono stati fittati con una Lorentziana di cui ci interessa il valore centrale, il quale indica la frequenza di risonanza. Valore stimato: 4.972GHz.

Il valore ottenuto va considerato una prima approssimazione: una tecnica più performante sarà descritta in seguito e si baserà su questo risultato.

Calibrazione di impulsi

Altro aspetto fondamentale per la caratterizzazione di un qubit è la calibrazione degli impulsi che implementano fisicamente i gate che si applicano in un circuito quantistico. Si pensi ai gate come le operazioni elementari del quantum computing.

In questa sezione vengono descritte le calibrazioni degli impulsi necessari per ottenere i gate π e $\pi/2$; nomi che si riferiscono alle ampiezze degli angoli di rotazione attorno all'asse delle X che questi gate apportano allo stato del nostro qubit (riferendosi alla sfera di Bloch). La procedura qui seguita è quella dell'esperimento noto in letteratura come *Rabi experiment* [16]; per la quale si scelgono a priori la forma dell'impulso e la sua durata: in questo caso sono stati scelti impulsi gaussiani di larghezza - ovvero la *sigma*- $0,075\mu s$ (mentre la frequenza utilizzata, chiaramente, è quella di risonanza trovata precedentemente). Si vuole specificare che, in questo esperimento, si è abbastanza liberi di scegliere la forma dell'impulso, sebbene per qubit superconduttori è preferibile scegliere una forma rappresentabile da una funzione il più possibile liscia (o *smooth*) [17] [18].

Si descrive ora l'esperimento. Al qubit in stato fondamentale vengono mandati in ingresso degli impulsi di durata costante e ampiezza via via crescente. Per ogni valore di ampiezza si inviano 1000 impulsi e si esegue una misura di energia al qubit. La frazione di volte che si ottiene $|1\rangle$ permette di identificare lo stato del qubit che in media si ottiene con questo valore di ampiezza. Si pensi di nuovo al qubit come un sistema a due livelli, il cui generico stato è rappresentabile come $\alpha|0\rangle + \beta|1\rangle$; considerando la condizione di normalizzazione $|\alpha|^2 + |\beta|^2 = 1$ possiamo ottenere i due coefficienti a meno della una fase relativa. Quello che si osserva, al variare dell'ampiezza degli impulsi inviata, sono delle oscillazioni periodiche proprio dei moduli quadrati di α e β . Questo fenomeno prende il nome di Oscillazioni di Rabi; la cui trattazione teorica è fornita in appendice, qui eliminata per semplicità espositiva.

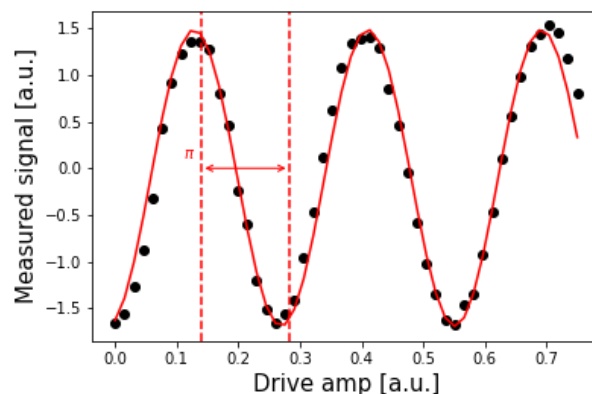


Fig 2.2 Le oscillazioni di Rabi

Dal fit di questi dati - fig 2.2 - con una sinusoide è possibile ottenere l'ampiezza necessaria per il gate π , rappresentata dal semiperiodo delle oscillazioni.

Forma funzionale usata:

$$A \cos\left(\frac{2\pi x}{T} - \varphi\right) + B$$

Dove A è una costante, x è il valore variabile dell'ampiezza dell'impulso inviato, T è il parametro rappresentante il periodo in ampiezza e sia φ -fase iniziale- che B sono altre due costanti

Si spiega ora il motivo per cui la durata dell'impulso sia stata tenuta costante. Si ricordi che l'obiettivo che ci si è posti è esclusivamente l'implementazione di un gate π accurato: l'importante è essere in grado di ruotare di un angolo π attorno all'asse delle X lo stato sulla sfera di Bloch e lasciar libero solo il parametro d'ampiezza è sufficiente per ottenere questo risultato. È pur vero che la durata del gate potrebbe avere ripercussioni per quanto interessa il quantum computing; per il quale, dati i tipici tempi di decoerenza di un sistema fisico (qubit), è molto vantaggioso avere gate brevi in durata in modo tale da poter implementare algoritmi che richiedano un numero maggiore di operazioni. Questo aspetto è discusso nel capitolo 3 come punto fondamentale del lavoro di ottimizzazione. Si è ora in grado di promuovere lo stato $|0\rangle$ allo stato eccitato $|1\rangle$ con ottima accuratezza.

Discriminatore di $|0\rangle$ e $|1\rangle$

Dal momento in cui si è in grado di implementare l'X gate (il nome che la letteratura dà al gate fin qui chiamato π) si intende ora mostrare come distinguere lo stato di ground e lo stato eccitato con una misura.

L'esperimento è assai elementare. Si creano due semplici circuiti quantistici: il primo è costituito da una sola operazione di misura del qubit, il secondo, invece, prevede di applicare l'X gate appena implementato e successivamente effettuare la misura. Si può quindi assumere che gli stati sui quali si effettuano le misure siano o $|0\rangle$ oppure $|1\rangle$ e non sovrapposizioni di questi.

In figura -fig 2.3- sono rappresentate le sequenze di impulsi inviati al qubit per i rispettivi circuiti.

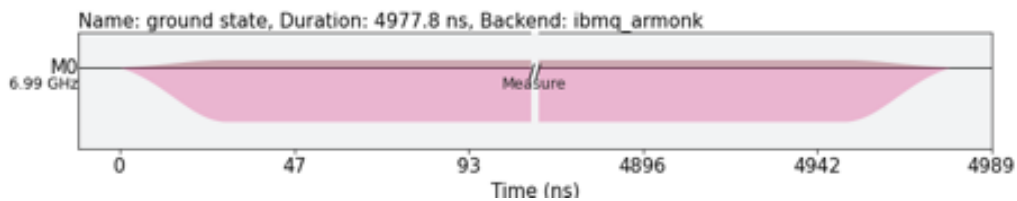


Fig 2.3: sola misura dello stato quantistico

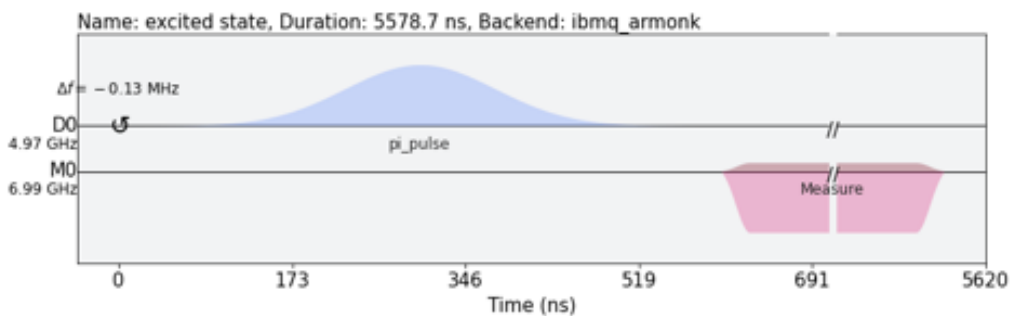
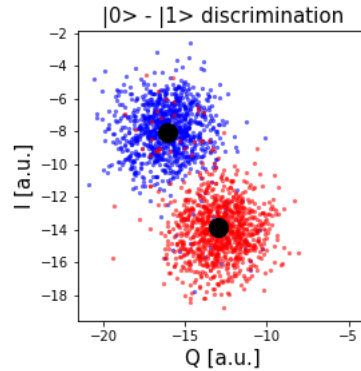


Fig 2.3: Applicazione di un impulso π seguito da immediata misura

I risultati delle misure, che anch'esse sono impulsi alla frequenza di risonanza del qubit, vengono interpretati come punti nel piano complesso (parte reale; parte immaginaria) con unità arbitrarie proprie delle librerie di Qiskit.



Si può facilmente notare come vadano a configurarsi due raggruppamenti ben definiti. I punti blu sono i risultati delle misure del programma di preparazione dello stato fondamentale (la sola misura) mentre i punti rossi sono quelli del programma con un X gate seguito da misura.

Da questi risultati si può costruire una funzione per discriminare i due stati. Si procede trovando il punto medio di uno e dell'altro cluster (evidenziati in nero nella figura), si dichiara una funzione che calcola la distanza di un generico punto da l'uno o dall'altro centro cosicché per programmi o circuiti successivi ogni risultato di misura sarà interpretato come $|0\rangle$ se il rispettivo punto sarà più vicino al punto medio del raggruppamento blu e $|1\rangle$ se altrimenti.

Si ricorda che in meccanica quantistica la misura è proiettiva, non si può dunque assolutamente pensare che i punti lontani dai "centri" possano essere interpretati come stati sovrapposti e che quindi ci possa essere un modo per ottenere informazioni sui coefficienti α e β dello stato in cui era il qubit prima della misura. Anzi, il fatto che si osservano due cluster, piuttosto che due soli punti nel piano, si deve al rumore o all'imperfezioni proprie del qubit e dell'apparato di misurazione.

Stima del tempo di rilassamento (T_1)

Il tempo di rilassamento di un qubit è un parametro fondamentale nella valutazione delle prestazioni dello stesso; in particolare, è importante che questo tempo caratteristico sia molto maggiore rispetto alla durata tipica di un'operazione sul qubit (gate). Comunemente chiamato anche T_1 , il tempo di rilassamento è definito come l'intervallo di tempo necessario affinché la probabilità di misurare $|1\rangle$ per un qubit preparato nel suo stato eccitato a $t=0$ si riduca da 1 a $1/e$. La definizione di T_1 è in completa analogia con la definizione di tempi di decadimento degli isotopi radioattivi (o delle eccitazioni atomiche) in quanto la natura del fenomeno descritto è esattamente la stessa: stocastica. Come l'istante in cui decade un isotopo è assolutamente imprevedibile, la stessa cosa vale per il passaggio dallo stato $|1\rangle$ allo stato $|0\rangle$ per un qubit se su questo non agisce alcun gate ed è, dunque, lasciato libero di interagire con l'ambiente. Si tratta in questo caso di esperimenti che richiedono molta statistica per poter ottenere delle stime affidabili.

Come si potrà poi constatare, il procedimento qui descritto utilizza tutto ciò che è stato implementato finora: l'X gate propriamente calibrato e la funzione *discriminator*. Di fatto, rispetto a quanto fatto finora c'è solo una novità: oltre ai gate (implementati con gli impulsi trovati) e all'operazione di

misura, ci avvaliamo anche di tempi di ritardo (*delay time*) tra gli uni e gli altri. Tempi che possiamo inserire nel circuito quantistico in maniera del tutto arbitraria grazie alle librerie di Qiskit.

L'esperimento prevede tre operazioni: applicare un π -pulse, inserire un ritardo e applicare una misura. Questo è stato iterato per 250 volte per ognuno dei 70 valori di delay time diversi (da $1\mu s$ a $450\mu s$ a step di $6,5\mu s$).

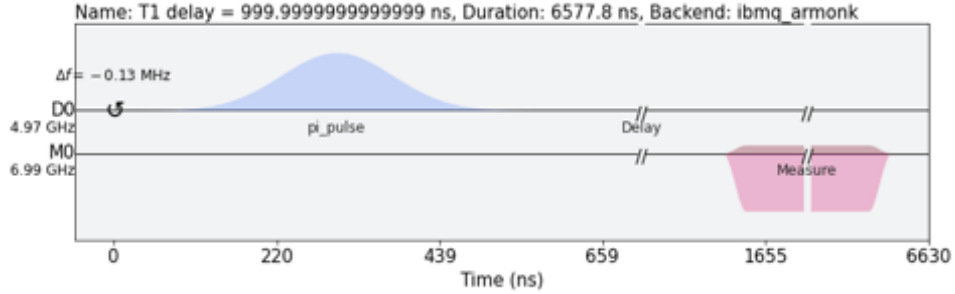


Fig 2.4: rappresentazione grafica dell'esperimento

Con questi dati si costruisce il grafico (fig 2.5) frazione di qubit nello stato $|1\rangle$ in funzione del delay inserito e si fittano i dati con un esponenziale decrescente della forma:

$$Ae^{-t/T_1} + B$$

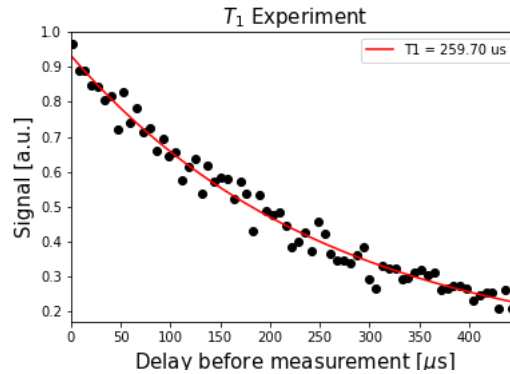


Fig 2.5: andamento della frazione di misure che hanno restituito $|1\rangle$ come risultato di ogni esperimento in funzione del tempo di ritardo inserito tra il gate X e la misura

Come si può notare dalla figura 2.5, la funzione esponenziale decrescente ben rappresenta la frazione di qubit che popola lo stato $|1\rangle$ in funzione del ritardo inserito; inoltre, il fit conferma quanto anticipato nel capitolo 1: i valori caratteristici di T_1 per qubit superconduttori sono dell'ordine delle centinaia di microsecondi.

Per completezza, si annota che durante il lavoro di tesi è capitato di osservare oscillazioni di T_1 tra i $150\mu s$ e i $260\mu s$; ad indicare l'enorme delicatezza di questi sistemi.

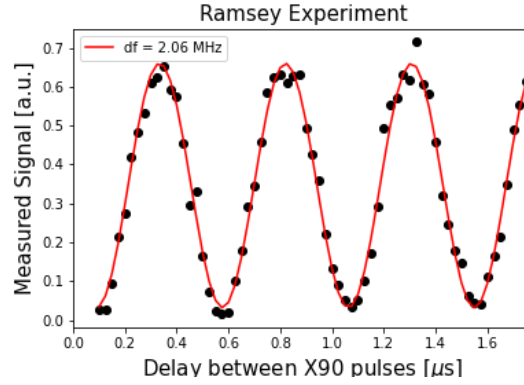
Ramsey Experiment

Come anticipato nel paragrafo "frequenza di risonanza", viene ora mostrato come stimare con maggior precisione tale frequenza. La procedura standard a tale fine prevede di modificare di qualche MHz la frequenza dell'impulso in ingresso al qubit (portandolo quindi fuori-risonanza, *detuning*) e con questa nuova frequenza inviare due impulsi $\pi/2$ intervallati da tempi di ritardo via via crescenti;

successivamente si ‘chiude il circuito’ con una misura. I risultati ottenuti mostrano un andamento sinusoidale in funzione della durata dei ritardi; questi dati vengono fittati con una funzione del tipo:

$$A \cos(2\pi Bx - C) + D$$

dove: A è l’ampiezza della sinusoide, B (df in figura) è il parametro il cui valore indica di quanto il nostro segnale è fuori risonanza col qubit, C è una fase iniziale, D un termine costante.



Così facendo, si è ora in grado di dare una stima più precisa della frequenza di risonanza semplicemente calcolando:

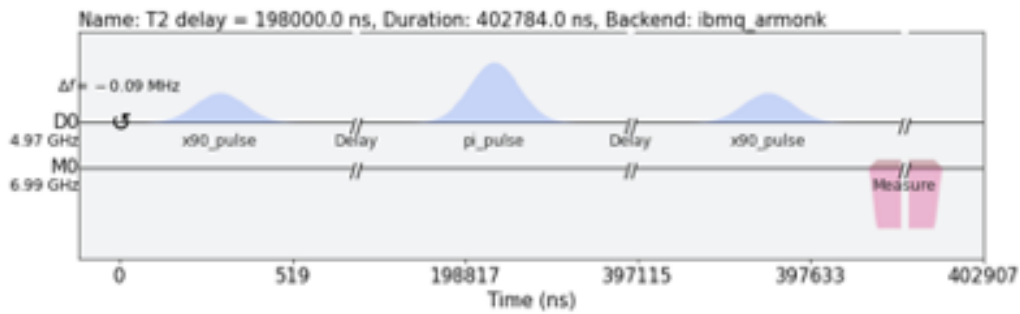
$$f_{precisa} = f_{approx} + detuning - B$$

Dove f_{approx} è la frequenza di risonanza stimata precedentemente, $detuning$ è il valore di discrepanza che si è aggiunto per mandare il segnale in ingresso fuori-risonanza, B è il parametro ottenuto dal fit.

Tempo di coerenza (T2)

Il tempo di coerenza è definito come il tempo per il quale lo stato quantistico mantiene inalterata la differenza di fase relativa tra $|0\rangle$ e $|1\rangle$, per questo si parla anche di rilassamento trasverso, mentre il rilassamento responsabile del T1 è detto rilassamento longitudinale) [1].

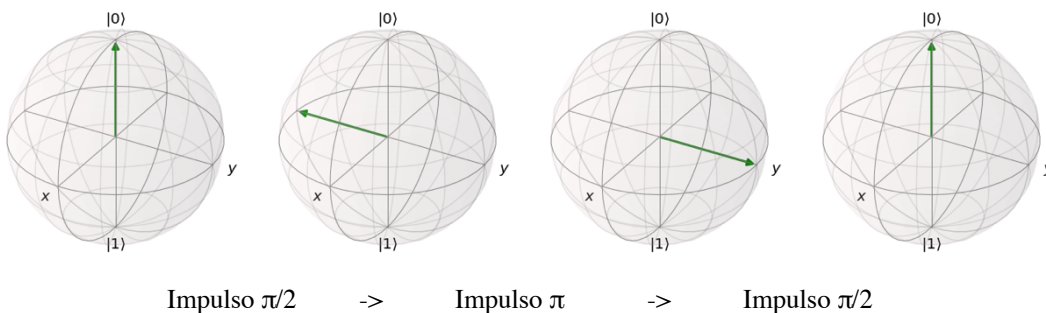
Per stimare il tempo di coerenza di un qubit si segue quello che in letteratura è detto “Hahn Echoes Experiment”; il quale si compone di una sequenza di impulsi $\pi/2 - \pi - \pi/2$ intervallati da un ritardo. Similmente all’esperimento Ramsey, si eseguono questi circuiti quantistici 500 volte per ogni ritardo e per 50 valori di ritardo diversi (da $4\mu s$ a $200\mu s$). Si vuol far notare che, preparato lo stato iniziale al livello fondamentale $|0\rangle$, la sequenza di impulsi porterebbe di nuovo allo stato $|0\rangle$ in quanto, composti tra loro, equivalgono ad una rotazione di un angolo 2π attorno all’asse X. I ritardi, dunque, sono necessari per permettere solo stato quantistico di evolvere e creare così una differenza di fase tra la sovrapposizione di stati $|0\rangle$ e $|1\rangle$; possiamo visualizzare questo fenomeno come la precessione dello stato del qubit sulla sfera di Bloch.



Esempio:

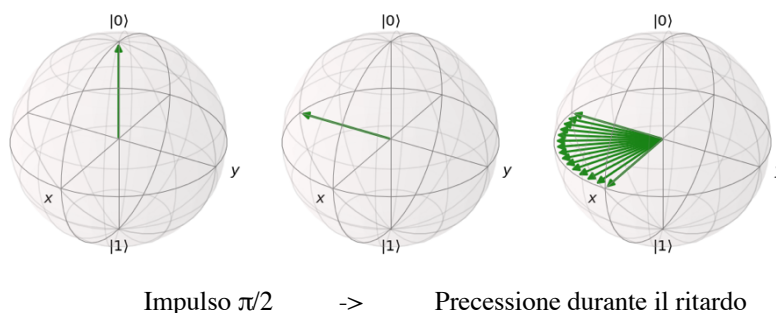
Si mostrano ora due diversi scenari: il caso in cui i diversi impulsi vengono inviati in sequenza immediata e il caso, durante il ritardo, si crea una differenza di fase relativa di $\pi/2$.

Caso 1: sequenza immediata



La sequenza illustrata rende evidente quanto sostenuto in precedenza: applicare i tre gate senza alcun ritardo tra di essi riporta lo stato quantistico allo stato fondamentale. Tutte e tre le rotazioni vanno intese in senso antiorario attorno all'asse delle X.

Caso 2: precessione di $\pi/2$ durante il ritardo



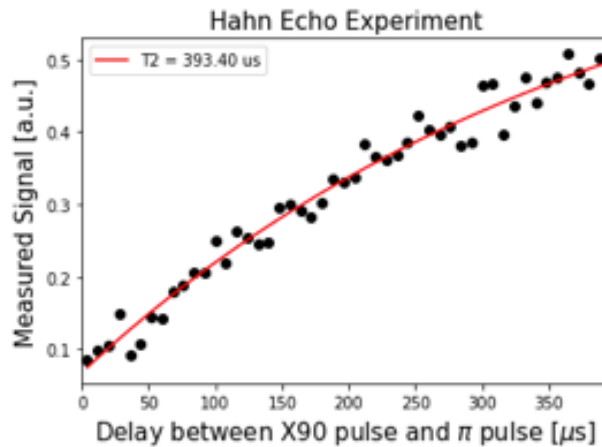
Si è mostrato ora un diverso scenario: dopo l'applicazione dell'impulso $\pi/2$ il vettore di stato ruota fino ad allinearsi con l'asse delle X, a questo punto una rotazione di angolo π attorno all'asse delle X sarebbe totalmente influente. Bisogna osservare che questo secondo scenario avverrebbe solo nel caso in cui fosse inserito un ritardo maggiore del T2; è un caso

non contemplato nel nostro esperimento ma è stato scelto come esempio dell'effetto della precessione per motivi di chiarezza espositiva.

Le misure effettuate a termine di ogni singolo esperimento permettono di creare il grafico che mostra l'andamento della frazione di volte in cui si ottiene $|1\rangle$ in funzione dei tempi di ritardo inseriti. Ci si aspetta, per quanto detto prima, che la curva del grafico abbia origine all'intersezioni degli assi e poi cresca; questa crescita è di tipo esponenziale, dunque si fittano i dati con una funzione del tipo

$$Ae^{-t/T_2} + B$$

Dove T_2 è il parametro di cui siamo interessati a conoscere il valore. Il valore ottenuto è la stima del tempo di coerenza.



Anche per questo parametro, durante il lavoro di tesi si sono ottenute stime molto diverse tra loro. Infatti, nel capitolo 1 è stato riportato il valore di T_2 pari a $223,56\mu$ s; il giorno in cui si è scritto questo paragrafo il valore riportato nella scheda tecnica di ibmq-armonk riportava $274,57\mu$ s.

Capitolo 3

Ottimizzare l'utilizzo di un qubit

Una volta imparato a manipolare un qubit (e.g. saper tradurre un gate logico in un preciso impulso), si deve pensare a come usarlo per il suo fine ultimo: essere utilizzato per il quantum computing. Anche questo aspetto è di fondamentale importanza, in quanto, come dimostrato nel capitolo precedente, un qubit ha dei limiti intrinseci, tra i quali gli errori di implementazione di singolo gate e i tempi di decoerenza. L'utilizzo ideale di un qubit deve ridurre al minimo queste cause di errore.

3.1 Riduzione di errori dovuti ai gate

Per ridurre la probabilità che degli errori scaturiscano a causa dei gate inseriti in un algoritmo, ci sono due strade percorribili: lavorare a livello di hardware o a livello di algoritmo stesso; in questo paragrafo si discute questa seconda.

Come prima soluzione di ottimizzazione si può sfruttare un po' di semplice algebra lineare: il prodotto tra matrici 2x2. Questo perché ogni gate è una trasformazione unitaria applicata ad un sistema a due livelli, dunque rappresentabile con una matrice che gode, per definizione, della proprietà:

$$U^\dagger U = \mathbb{I}$$

Ragionando in termini “*gate=matrice*” si capisce come applicare più gate ad un qubit (vettore-stato di dimensione 2) equivalga a moltiplicare un vettore per un certo numero di matrici 2x2. Il risultato sarà comunque una singola trasformazione unitaria; infatti il prodotto di due matrici unitarie è una matrice unitaria. Inoltre, il prodotto tra matrici gode della proprietà associativa:

$$AB v = (AB)v$$

dove A e B sono matrici e v un vettore.

Analogamente si possono fare composizioni tra i gate (matrici unitarie) applicati ad un qubit (vettore-stato). Il risultato sarà un unico gate implementabile in un tempo minore rispetto a quello necessario per implementare tutti i gate originari: un gate U3 con opportuni parametri. Schematicamente:

$$G_n G_{n-1} \dots G_2 G_1 |\psi\rangle \rightarrow U3(\theta, \varphi, \lambda) |\psi\rangle$$

Dove G_n rappresenta il gate *n-esimo* applicato al qubit.

Per vedere come la riduzione di numero di gate comporti un vantaggio in termini di costo computazionale si riporta un esempio costruito utilizzando simulatori di computer quantistici.

Esempio:

Creo una generica sequenza (*seqI*) di gate i già noti X, Rz, P e Ry da applicare ad un qubit. Di seguito una rappresentazione (fig. 3.1).

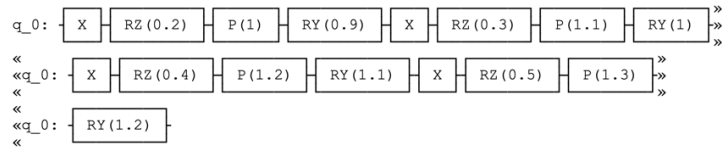
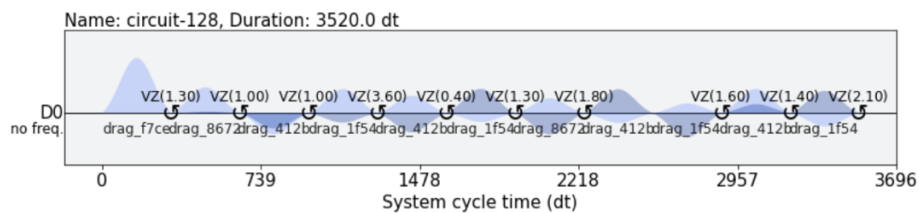
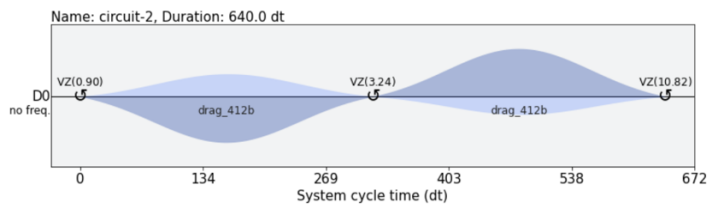


Fig. 3.1: Una sequenza arbitraria di gate applicati al qubit q_0

Utilizzando un simulatore (in questo caso *FakeArmonk*, in dotazione con le librerie Qiskit) mostro gli impulsi necessari per implementare questa sequenza:



Di questo grafico osservo un unico dato: la durata, in unità arbitrarie, pari a 3520dt. Ora mostro gli impulsi necessari nel caso in cui la sequenza sia ridotta ad un unico U3 gate:



Ora la durata è di 640dt! La cosa interessante da osservare è che questa durata non dipende in alcun modo dalla lunghezza della sequenza originale e questo perché il gate U3 viene realizzato con al più tre rotazioni; che sono tre “operazioni” elementari che richiedono un unico impulso ciascuna.

Con l’esempio si è mostrato un caso in cui la riduzione di tempo di esecuzione è di circa un fattore 1/6; questo risultato, però, merita una precisazione. È stata ottenuta una riduzione notevole in virtù del fatto che la sequenza trasformata in singolo gate fosse particolarmente lunga. Di fatto, nella quasi totalità degli algoritmi vengono utilizzati più di un qubit e, in genere, non capita che si debbano applicare molti gate su un singolo qubit senza che questo venga fatto interagire con un altro; dunque, ci si può aspettare dei risultati più modesti da questa riduzione. Per chiarezza, si osservi la fig 3.2. rappresentante un circuito più “plausibile” di quello dell’esempio precedente.

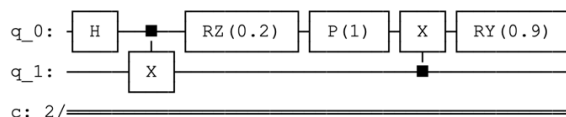


Fig 3.2: Nel circuito mostrato gli unici gate che possono essere composti sono

Rz e P dato che i gate CNOT (indicati con una X e un segmento tra i due canali) implicano l'interazione tra qubit diversi.

Nelle librerie di Qiskit le composizioni tra gate su singolo qubit sono calcolate in automatico con il metodo *transpile()*.

```
compiled_circuits = transpile(qc, backend=backend)
```

Questa semplice linea di codice restituisce un circuito compilato in cui ogni sequenza di gate su singolo qubit è ridotta ad un unico U3 prendendo per argomento il circuito quantistico corrispondente ad un certo algoritmo e il backend sul quale si intende eseguire il circuito. Per completezza, è necessario specificare che il metodo *transpile()* faccia in realtà molto di più, in quanto adegua un algoritmo ad un determinato backend a seconda delle caratteristiche di ogni suo singolo qubit e della sua architettura interna; ma questo è un argomento che esula da quanto trattato nel capitolo.

3.2 Utilizzare gate appena calibrati

Una volta minimizzato il numero di gate necessari per svolgere un determinato algoritmo quantistico, è bene massimizzare l'affidabilità di ogni singolo gate.

Per quest'ulteriore tecnica di ottimizzazione, non è necessario introdurre nulla di veramente nuovo: bisogna solo rifarsi al lavoro di calibrazione del capitolo 2. Si tratta, infatti, di sfruttare quanto fatto a livello hardware per avere vantaggi a livello di quantum computing. L'idea di fondo di questo capitolo è che i computer quantistici sono strumenti estremamente delicati e il loro utilizzo è fruttuoso solo a seguito di una rigorosa calibrazione. Se non ci si volesse basare sulle calibrazioni giornaliere fatte da IBM (che, di conseguenza, hanno al più qualche ora di "età"), si può fare affidamento ad altre ancora più recenti: le nostre. Sì, perché quanto fatto su QiskitPulse può essere facilmente utilizzato in Qiskit. Ad esempio, la linea di codice necessaria per aggiungere un X gate al primo qubit di un *QuantumCircuit* chiamato qc è:

```
qc.x(0)
```

In questo modo non si hanno informazioni sull'impulso di microonde inviato al qubit, si utilizza quello implementato con l'ultima calibrazione fatta al computer. Per usare, invece, l'impulso coi parametri appena ottimizzati si devono usare i seguenti metodi:

```
with pulse.build(backend) as pi_pulse:
    drive_duration = ...
    get_closest_multiple_of_16(pulse.seconds_to_samples(drive_duration_sec))
    drive_sigma = pulse.seconds_to_samples(drive_sigma_sec)
    drive_chan = pulse.drive_channel(qubit)
    pulse.play(pulse.Gaussian(duration=drive_duration,
                              amp=pi_amp,
                              sigma=drive_sigma,
                              name='pi_pulse'), drive_chan)
qc.add_calibration('x', [0], pi_pulse)
```

In cui durata, sigma e ampiezza dell'impulso gaussiano sono i parametri ottenuti con le procedure spiegate nel capitolo 2. Così facendo, si controlla pienamente la corrispondenza gate $\leftarrow \rightarrow$ impulso. Dalla chiamata del metodo `qc.add_calibration()` il programma smetterà di usare il gate pre-impostato ed inizierà a usare l'impulso creato dall'utente.

Per avere un confronto tra i diversi approcci è stato eseguito un esperimento. Sempre utilizzando il backend `ibmq-armonk`, si sono eseguiti dei circuiti elementari (applicazione di un X gate e misura del qubit) in tre modi differenti:

1. Utilizzando le librerie Qiskit. Si tratta semplicemente di dichiarare un oggetto della classe `QuantumCircuit`, applicare un X gate seguito poi da una misura. Il risultato atteso di questo circuito è, ovviamente, lo stato $|1\rangle$, per stimare l'affidabilità di questo gate preimpostato si è commissionato al servizio cloud di IBM l'esecuzione di questo algoritmo 30 volte con 1000 tentativi ciascuno per ottenere sufficiente statistica.
2. Utilizzando le librerie QiskitPulse. In questo secondo caso si è ripercorso il programma di caratterizzazione fino alla dichiarazione del π -pulse; giunti a questo punto non si è proceduto con la stima di T_1 e T_2 ma si è costruito il medesimo circuito descritto nel punto 1; in questo, però, l'X gate è stato implementato con l'impulso appena calibrato. Il numero di esecuzioni richieste è pari a quelle precedenti.
3. Utilizzando le librerie QiskitPulse ma con un lavoro di calibrazione più completo. Come anticipato nel Capitolo 2, è possibile affinare la calibrazione del π -pulse iterando il processo. Di fatti, si è mostrato come costruire questo impulso utilizzando solo una stima approssimativa della frequenza di risonanza; questo perché si è immaginato di non esser ancora in grado di passare da $|0\rangle$ a $|1\rangle$, il che sarebbe poi stato necessario per i vari esperimenti come il Rabi e il Ramsey. Ma, seguendo l'esempio del paper [16], ai fini di una migliore ottimizzazione dei parametri di questo impulso è consigliato seguire i seguenti step:
 - Trovare una prima stima di frequenza di risonanza utilizzando tecniche di spettroscopia all'output del qubit; come fatto nel paragrafo "Frequenza di risonanza" del Capitolo 2.
 - Impulsare il qubit a questa frequenza in modo da osservare le oscillazioni Rabi e ottenere da un fit con una funzione sinusoidale l'ampiezza necessaria per implementare un π -pulse.
 - Eseguire un esperimento di Ramsey in modo da ottenere una stima più precisa della frequenza di risonanza.
 - Eseguire un ulteriore esperimento per osservare le oscillazioni Rabi ed estrarre da un nuovo fit l'ampiezza definitiva dell'impulso necessario per passare dallo stato $|0\rangle$ allo stato $|1\rangle$.
 Anche per questo terzo metodo sono stati eseguiti 30 volte set di 1000 circuiti ripetuti.

In tabella di riportano i risultati dei tre esperimenti; in particolare si mostrano i valori medi del numero di volte in cui si è ottenuto $|1\rangle$ su un set di 1000 esecuzioni.

Metodo 1	Metodo 2	Metodo 3
904 ± 2	903 ± 2	904 ± 2

Si reputano questi risultati estremamente indicativi delle prestazioni del qubit e della bontà delle calibrazioni giornaliere di IBM. La prima cosa che si vuole osservare è che si ottiene $|1\rangle$ come risultato della misura solo nel 90% dei casi (circa). Questo è piuttosto sorprendente. Di fatti si è considerato il circuito più semplice da eseguire, in cui compare un unico gate che non ha nemmeno

bisogno di essere scomposto in operazioni più elementari; ci si poteva aspettare un valore decisamente più prossimo a 100.

In seconda battuta, si vuole commentare la perfetta compatibilità dei tre metodi utilizzati. L'unico a sembrare un po' meno performante rispetto agli altri sembra essere quello che si basa sulla stima "grezza" della frequenza di risonanza. C'è però un dettaglio che si vorrebbe rimarcare: per svolgere questa tesi non si ha avuto a completa disposizione un qubit, tutti i circuiti inviati al servizio cloud sono stati messi "in coda" con altri di altri utenti e questo va a discapito delle calibrazioni fatte con QiskitPulse e poi utilizzate nei circuiti. Il caso ideale, che si è descritto a inizio paragrafo, prevede che le esecuzioni dei circuiti siano immediatamente successive alle calibrazioni, il che potrebbe portare una maggiore affidabilità del metodo 3.

Altro aspetto non trascurabile per il commento dei risultati è l'errore nella misura dello stato di un qubit. Nella scheda tecnica del computer ibmq-armonk si legge "*Avg. Readout Error: 3.86e-2*". Il che significa che l'errore di lettura dello stato di un qubit è affetto da un'incertezza relativa nell'ordine del 4%! Il che è ben al disopra delle deviazioni standard dalla media riportate in tabella. Per poter dire in che misura i risultati ottenuti siano dovuti all'errata implementazione del gate o all'errore di lettura dello stato, bisognerebbe condurre ulteriori approfondimenti. Questi, però, esulano dallo scopo di questo lavoro interessato al confronto tra gate preimpostati e gate creati dall'utente; i quali sono stati qui confrontati a parità di hardware, dunque a parità di apparato condizioni.

3.3 As Late As Possible (ALAP)

"*As Late As Possible*" è l'approccio alla schedulazione utilizzato di default da Qiskit. Sebbene si possa utilizzare il suo opposto "*As Soon As Possible*", c'è un ottimo motivo fisico per scegliere l'approccio ALAP: la decoerenza. L'idea di base è quella che un qubit lasciato allo stato fondamentale ha poche probabilità di cambiare di stato, mentre un qubit in una sovrapposizione di stati o esattamente nello stato $|1\rangle$ è soggetto sia a rilassamento trasverso che longitudinale.

Chiaramente, questo discorso è valido esclusivamente per circuiti in cui è presente un numero plurale di qubit, in cui è possibile che solo un certo sottogruppo venga utilizzato nella prima parte di algoritmo e solo più tardi vengano "chiamati in causa" i restanti.

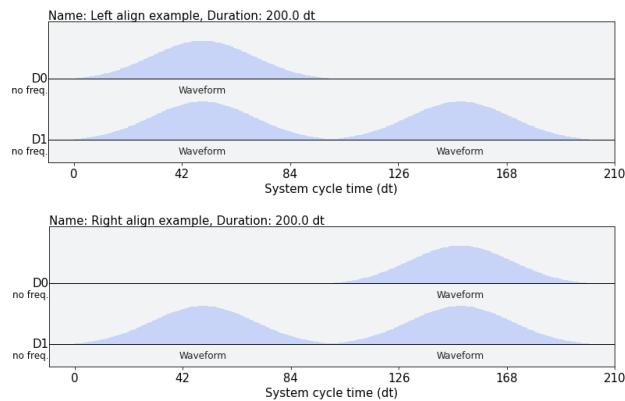


Fig. 3.2: In figura sono rappresentati i due diversi approcci di schedulazione ALAP e ASAP. Importante notare come nel caso destro (ALAP) nessun qubit venga lasciato interagire liberamente con l'ambiente dopo il primo impulso.

