Tomographie semi-supervisée de processus quantiques unitaires

Francois Verdeil

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *20/12/2023* **par :**

François VERDEIL

Tomographie semi-supervisée de processus quantiques unitaires

---

JURY

Ali MANSOUR	ENSTA Bretagne	Président du Jury
Gilles BUREL	Université de Bretagne Occidentale	Rapporteur
François CHAPEAU-BLONDEAU	Université d'Angers	Rapporteur
Claire GOURSAUD	INSA Lyon	Examinatrice
Yannick DEVILLE	Université Paul Sabatier	Directeur de thèse

École doctorale et spécialité :
MITT : Signal, Image, Acoustique et Optimisation
Unité de Recherche :
IRAP
Directeur(s) de Thèse :
Yannick DEVILLE
Rapporteurs :
Gilles BUREL et *François CHAPEAU-BLONDEAU*

Tomographie semi-supervisée de processus quantiques unitaires

François Verdeil

2023

Remerciements

Je tiens à exprimer ma plus profonde gratitude envers Yannick Deville, mon directeur de thèse, pour sa guidance exceptionnelle tout au long de ce parcours académique. Sa patience infinie face à mes mauvaises formulations et mes fautes de frappes en série (surtout les versions en mode pluriel) et son soutien inconditionnel m'ont permis d'explorer des pistes nouvelles et d'évoluer dans un environnement propice à la créativité scientifique. Je réalise que l'encardrement de la thèse, particulièrement les réunions au début, et les relectures d'articles ensuite, lui ont pris beaucoup de temps, et je lui en suis très reconnaissant.

Mes remerciements vont également à Gilles Burel et François Chapeau-Blondeau, ainsi qu'à Ali Mansour et Claire Goursaud, pour leur investissement dans la lecture minutieuse du manuscrit, leurs retours constructifs et leurs encouragements le jour de la soutenance.

Enfin, un immense merci à mes collègues de bureau qui ont été des compagnons de route précieux : Hervé Carfantan, pour nos discussions enrichissantes autour d'une tasse de café et sa bonne humeur contagieuse, Mehdi, pour son soutien inconditionnel et son amitié sincère dès les débuts de cette thèse, ainsi que Mostafa et Tin, pour leurs contributions précieuses et leur compagnie agréable qui ont rendu cette expérience plus riche et Shahram, pour son rôle crucial dans le premier entretien à l'IRAP et sa gestion remarquable des travaux pratiques.

Leur présence a été d'une valeur inestimable, rendant cette expérience de thèse à la fois productive sur le plan académique et humainement gratifiante.

Résumé

L'objectif principal de cette thèse est de développer des algorithmes de tomographie de processus quantiques. Ce problème est étudié depuis la fin des années 1990 dans la littérature car les portes quantiques sont les blocs de base de la plupart des ordinateurs quantiques, et estimer leurs paramètres est nécessaire pour réaliser des portes de meilleure qualité. Le nombre de paramètres réels d'un processus quelconque est $2^{4n_{qb}} - 2^{2n_{qb}}$ (n_{qb} est le nombre de bits quantiques ou qubits), ce nombre devient très rapidement prohibitif (240 paramètres pour deux qubits, 4032 paramètres pour trois qubits). Afin de s'affranchir de ce problème, nous supposons que le processus à étudier est unitaire, ce qui est garanti dans un système fermé. Le nombre de paramètres d'un système unitaire dépend toujours exponentiellement du nombre de qubits mais de façon plus raisonnable : $2^{2n_{qb}}$ (16 pour deux qubits, 64 pour trois qubits). La méthode de tomographie de processus que nous proposons fonctionne avec des états d'entrée quelconques (ou presque), on suppose seulement qu'il est possible de préparer plusieurs copies d'un ensemble initialement inconnu d'états purs d'entrée et d'en mesurer avant et après que le processus à estimer ait été appliqué. Après avoir estimé les états et levé des indéterminations sur des paramètres à estimer, cet algorithme se résume à un problème de moindres carrés linéaires avec contrainte d'unitarité. Ce problème peut être résolu analytiquement et sans point initial, il est donc possible d'identifier une porte sans aucune connaissance préalable. Pour avoir une estimation plus précise des paramètres de la porte, un algorithme de maximum de vraisemblance (plus lent et nécessitant un bon point initial fourni par la version de base de l'algorithme) est proposé.

Nos algorithmes de tomographie de processus fonctionnent avec tout ensemble de types de mesures qui permet d'identifier les états, mais nous proposons nos types de mesures et algorithmes d'estimation d'états adaptés. Nous proposons aussi un algorithme qui permet d'identifier certains paramètres des mesures quantiques que nous utilisons et de faire la tomographie de processus en une seule expérience. Ainsi, nous pouvons envisager d'identifier un processus avec une précision qui ne dépendra ni de la précision avec laquelle on prépare des états d'entrée de référence (car notre algorithme fonctionne avec des états quelconques, et que tous les états qui nous intéressent sont mesurés et estimés), ni de la connaissance a priori que l'on a sur les mesures quantiques réalisées.

Mots clés : Tomographie de processus quantique, Tomographie d'état quantique, Tomographie de mesure quantique, Récupération de phase, Problème de Procrustes, Maximum de Vraisemblance

Abstract

The main objective of this thesis is to develop quantum process tomography algorithms. This problem has been studied since the late 1990s in the literature because quantum gates are the building blocks of most quantum computers, and estimating their parameters is necessary to make better gates. The number of real parameters for any process is $2^{4n_{qb}} - 2^{2n_{qb}}$ (n_{qb} is the number of quantum bits or qubits), this number quickly becomes prohibitive (240 parameters for two qubits, 4032 parameters for three qubits). To avoid this problem, we assume that the process to be studied is unitary, which is always the case in a closed system. The number of parameters for a unitary process still depends exponentially on the number of qubits, but more reasonably: $2^{2n_{qb}}$ (16 for two qubits, 64 for three qubits). The process tomography method we propose works with almost any input states, we just assume that it is possible to prepare several copies of an initially unknown set of pure input states and measure them before and after the process to be estimated has been applied. From this system, a process tomography algorithm based on pure state tomography is proposed. Once indeterminacies have been lifted, this algorithm boils down to a linear least squares problem with a unitarity constraint. This problem can be solved analytically and without an initial point, so it is possible to identify a gate without any prior knowledge. To get a more accurate estimate of the gate parameters, a maximum likelihood algorithm (which is slower and requires a good initial point provided by the basic version of the algorithm) is proposed.

Our process tomography algorithms work with any set of measurement types that identifies the states, but we propose our own measurement types and state estimation algorithms. We also propose an algorithm that can identify some of the parameters of the quantum measurements we use and perform process tomography with a single experiment. Therefore, we can hope to identify a process with an accuracy that will depend neither on the precision with which we prepare reference input states (since our algorithm works with any states, and all the states of interest are measured and estimated), nor on the prior knowledge we have of the quantum measurements we perform.

Keywords : Quantum process tomography, Quantum state tomography, Quantum measurement tomography, Phase recovery, Procrustes problem, Maximum likelihood

Table des matières

Notations et acronymes

Afin d'accompagner le lecteur dans la compréhension du manuscrit, ce chapitre détaille les acronymes et les notations les plus utilisés.

Acronymes

AAQPT	"Ancilla-Assisted Quantum Process Tomography".
BFGS	"Broyden–Fletcher–Goldfarb–Shanno".
BQPT	"Blind Quantum Process Tomography".
fdr	"fonction de répartition".
GST	"Gate Set Tomography".
iid	indépendant.e.s identiquement distribué.e.s.
ML	"Maximum Likelihood".
LS	"Least Square".
POVM	"Positive Operator-Valued Measurement".
QMT	"Quantum Measurement Tomography".
QPT	"Quantum Process Tomography".
QST	"Quantum State Tomography".
SGQT	"Self Guided Quantum Tomography".
SQPT	"Standard Quantum Process Tomography".
std	"standard deviation" (écart-type).

Objets et opérations mathématiques

$\mathbb{R}$	Ensemble des réels.
$\mathbb{C}$	Ensemble des complexes.
$\mathbb{U}_d$	Ensemble des matrices unitaires de taille d.
$\mathbb{H}_d$, $\mathbb{H}_d^+$	Ensemble des matrices hermitiennes (resp. hermitiennes positives) de taille d.
$\otimes$	Produit tensoriel.
$. \odot .$	Produit terme à terme.
$.^*$	Trans-conjuguée.
$.^T$	Conjuguée.
$\underline{.}$	Transformation qui enlève le dernier élément d'un vecteur ou la dernière ligne d'une matrice.
$\lvert.\rvert, \lvert.\rvert^2$	Module et module au carré, terme à terme si appliqués à un vecteur.
$(.)_j$	k-ième élément d'un vecteur.
$(.)_{(j,k)}$	Elément en ligne j et colonne k d'une matrice.
$\boldsymbol{\delta}_j$	Vecteur (on doit préciser la taille) qui ne contient que des 0 sauf un 1 en position j.
$\delta_{j,k}$	Scalaire, vaut 1 si $j = k$, 0 sinon.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$	Loi gaussienne de moyenne $\boldsymbol{\mu}$ et covariance $\boldsymbol{\Gamma}$.

Par souci de clarté, nous avons noté tous les vecteurs en minuscule et gras, et toutes les matrices en majuscule et gras. Les seules exceptions sont les matrices densité des états mélanges $\boldsymbol{\rho}$, et

la matrice de processus d'un processus quantique quelconque (pas forcément unitaire), $\boldsymbol{\chi}$. Nous notons ces matrices en minuscule pour être cohérent avec la littérature.

Notations récurrentes

$\mathbf{A}$	Concaténation des matrices de vecteurs propres (voir, e.g. section 1.2.2).
$\mathbf{B}_{j,k}$	Matrice de taille d avec d^2 zéros et un 1 en position j, k.
$\mathbf{E}_{\mathcal{M}}$	Matrice de vecteurs propres associée au type de mesure $\mathcal{M}$ (pas toujours précisé) définie dans la Section 1.1.4.
f	Fidélité, définie en section 1.1.9.
$\boldsymbol{e}_1, ..., \boldsymbol{e}_d$	Base de référence de l'espace de Hilbert. Par défaut $\boldsymbol{e}_j = \boldsymbol{\delta}_j \ \ \forall j$.
$\mathbf{H}_{\mathcal{M}}$	Matrice hermitienne associée à $\mathcal{M}$ définie dans la Section 1.1.4 (peu utilisée).
d	Dimension de l'espace de Hilbert $d = 2^{n_{qb}}$.
$\mathbf{I}_d$	Matrice identité de taille d.
$\mathbf{K}_j$	j-ième opérateur de Kraus (1.7).
$\mathcal{L}_{exact}$	Vraisemblance de mesures contenues dans $\boldsymbol{n}$ (section 2.4.2) ou $\mathbf{N}$ (section 3.3.2) issue d'une loi multinomuiale .
$\mathcal{L}_{gauss}$	Version gaussienne régularisée de la vraisemblance définie pour $\boldsymbol{n}$ (section 2.4.3) ou $\mathbf{N}$ (section 3.3.2).
$\mathbf{M}$	Matrice de $\mathbb{U}_d$ à identifier, elle représente le processus unitaire.
$\widehat{\mathbf{M}}_{LS}, \widehat{\mathbf{M}}_{ML}$	Estimées de $\mathbf{M}$ des algorithmes de QPT des sections 3.1 et 3.3.
$\mathcal{M}$	Un type de mesure quantique. S'il y a des X, Y, Z en indice de $\mathcal{M}$, alors il s'agit des types de mesures de 2.1.1.
$\boldsymbol{n}$	Vecteur de taille dn_t qui contient les nombres d'occurrences de chacun des d résultats possibles des n_t types de mesures (répétées n_c fois) pour un seul état.
$\mathbf{N}$	Matrice $(dn_t) \times (n_i n_s)$ qui contient les nombres d'occurrences de chacun des d résultats possibles de tous les types de mesures (répétées n_c fois) sur tous les états mesurés.
n_{qb}	Nombre de qubits.
n_t	Nombre de types de mesures effectuées (applicable pour la QST et la QPT).
n_i	Nombre d'états initiaux utilisés (QPT).
n_s	Nombre d'instant auxquels au moins une mesure est réalisée (QPT).
n_c	Nombre de répétitions de chaque type de mesure sur chaque état mesuré (QST et QPT).
$\widehat{\boldsymbol{v}}_{pc}, \widehat{\boldsymbol{v}}_{rec}, \widehat{\boldsymbol{v}}_{ML}$	Estimées des vecteurs d'états avec les algorithmes de QST des sections 2.2, 2.3 et 2.4 respectivement.
$\mathbf{X}$	États d'entrée de la configuration virtuelle de la Fig. 3.3, défini dans (3.4).
$\mathbf{Y}$	États de sortie de la configuration virtuelle de la Fig. 3.3, défini dans (3.4).
$\boldsymbol{v}$	Vecteur d'état qui représente un état pur.
Δ_t	Intervalle de temps qui est tel que le système considéré change de $\boldsymbol{v}$ à $\mathbf{M}\boldsymbol{v}$.
ϵ	Processus quantique quelconque (pas forcément unitaire).
$\boldsymbol{\varepsilon}$	Vecteur des différences entre les probabilités théoriques et empiriques d'occurrence de chaque résultat de mesure. Défini pour la première fois en section 2.4.3.
$\boldsymbol{\rho}$	Matrice densité d'un état mélange, voir section 1.1.8.
$\boldsymbol{\chi}$	Matrice de processus d'un processus quelconque, voir (1.8).

Introduction

L'informatique quantique, imaginée dans les années 1980, est une technologie prometteuse pour laquelle beaucoup d'applications sont envisagées, mais qui se heurte à des contraintes physiques qui font que les applications les plus intéressantes sont encore hors de portée.

Alors que les premières architectures d'ordinateurs quantiques émergeaient en 2000, D. DiVincenzo a proposé cinq conditions [DiV00] pour que les ordinateurs quantiques puissent résoudre des problèmes utiles avec les algorithmes qui avaient été imaginés dans les décennies précédentes. Les cinq conditions sont les suivantes :

1. Un système physique "extensible" de qubits bien connus.
2. La possibilité d'initialiser les bits quantiques (ou qubits) en un état connu de référence (en général $|0...0\rangle$).
3. Des temps de cohérence assez longs pour réaliser de nombreuses opérations.
4. Un ensemble universel de portes quantiques bien réalisées.
5. La possibilité de mesurer les états des qubits indépendamment.

Dans le présent manuscrit, nous nous concentrons sur la condition 4. La qualité des portes quantiques limite fortement les performances des ordinateurs quantiques actuels, en effet (i) les ordinateurs quantiques à ions piégés ont des temps de cohérences de plusieurs secondes, ce qui est largement supérieur au temps de calcul par les portes (condition 3 satisfaite), et (ii) les "SPAM errors" (qui caractérisent les erreurs de préparation et mesure des états) sont en général beaucoup plus faibles que les erreurs des portes d'intrication multi-qubit[1] (les conditions 2 et 5 sont donc mieux satisfaites que la condition 4).

Un outil très important pour la réalisation de portes quantiques fidèles est la tomographie de processus quantiques (ou QPT pour "quantum process tomography"). En effet, les portes quantiques ne sont qu'une vue de l'esprit, elles sont généralement réalisées physiquement en appliquant un processus quantique (généralement unitaire et réalisé avec un hamiltonien constant) à l'état d'entrée. La QPT vise à identifier les paramètres d'un processus quantique. Elle permet donc de vérifier que la porte associée correspond bien à la porte que l'on voulait réaliser, si ce n'est pas le cas, on sait comment corriger le processus. En général, la QPT est réalisée en préparant des états initiaux prédéterminés auxquels on applique le processus, puis on mesure la sortie de façon à identifier les états de sortie, on peut ensuite calculer les paramètres du processus. Cette approche repose sur la tomographie d'état (ou QST pour "quantum state tomography"), qui a pour objectif d'identifier un état (en l'occurrence, il s'agit des états de sortie) à partir de résultats de mesures quantiques réalisées sur cet état. Pour ce faire, il faut de nombreuses copies de l'état à identifier car un état ne peut être mesuré qu'une fois et une seule réalisation d'une mesure quantique ne suffit pas à identifier un état en général.

[1] Voir les performances (fidélité SPAM et fidélité des portes à 2 qubits) de l'ordinateur que nous utilisons en section 5.2.

Un problème important qui limite les performances des algorithmes de QPT disponible dans la littérature est que les états d'entrée doivent être bien préparés, et la précision de l'estimée de la porte que donne la QPT ne peut pas être meilleure que la précision des états d'entrée préparés. Or ces états sont préparés avec des portes quantiques, et le fait d'avoir besoin de portes quantiques bien caractérisées pour identifier une porte quantique est problématique. Les erreurs de préparation des états d'entrée sont appelées "erreurs systématiques" car elles sont les mêmes sur toutes les copies des états (il existe aussi une erreur "centrée" qui est différente sur toutes les copies des états mais on la néglige car on considère que la préparation des états d'entrée est répétable et bien répétée pour chaque copie).

Par rapport aux algorithmes de QPT de la littérature, les algorithmes proposés dans le présent manuscrit ont les spécificités suivantes :

- nous considérons que le processus à identifier est unitaire. Cela rend nos méthodes moins générales, mais réduit considérablement le nombre de paramètres à estimer.
- Les valeurs que doivent prendre les états d'entrée ne sont pas imposées par les méthodes de QPT.

Le fait que l'on ne contraigne pas les valeurs prises par les états d'entrée fait que l'on peut utiliser des configurations (que nous allons expliquer plus en détail) qui sont équivalentes à la configuration de la figure 1 :

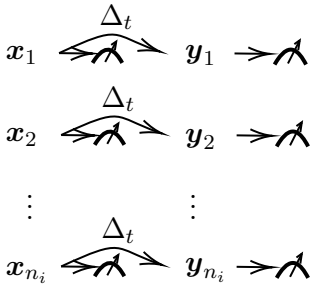


FIGURE 1 : protocole de QPT basique, les "double-flèches" signifient que la moitié des états sont mesurés, et l'autre moitié est modifiée par le processus unitaire à identifier. Les $\boldsymbol{x}_j$ représentent les états d'entrée, on les soumet à un hamiltonien constant pendant Δ_t pour leur appliquer le processus à identifier. Les $\boldsymbol{y}_j$ représentent les états en sortie du processus.

Des copies des états $\{\boldsymbol{x}_j\}_j$ en entrée du processus sont préparées, la moitié de ces copies est mesurée pour estimer les états $\{\boldsymbol{x}_j\}_j$ l'autre moitié des copies est modifiée par le processus à identifier (les $\{\boldsymbol{x}_j\}_j$ deviennent les $\{\boldsymbol{y}_j\}_j$) puis mesurée pour estimer les états $\{\boldsymbol{y}_j\}_j$. On en déduit les paramètres du processus qui a transformé les $\{\boldsymbol{x}_j\}_j$ en les $\{\boldsymbol{y}_j\}_j$.

L'avantage de cette configuration est que l'on supprime totalement le problème des erreurs systématiques mentionné ci-dessus (les états d'entrée $\{\boldsymbol{x}_j\}_j$ sont estimés et non prédéterminés, ils ne peuvent donc pas être mal préparés). Cette idée de faire des mesures à plusieurs niveaux n'est pas si complexe. Elle est présente dans la littérature (sous des formes différentes), mais pas pour la QPT adaptée aux processus unitaires. En effet, les algorithmes de QPT adaptés aux processus unitaires imposent les valeurs que doivent prendre les états d'entrée. Ce n'est pas le cas avec notre algorithme qui identifie le processus quels que soient les états d'entrée (en pratique il existe des conditions très peu contraignantes sur les états d'entrée, mais nous montrons que notre algorithme fonctionne très bien avec des états d'entrée aléatoires).

Le présent manuscrit contient cinq chapitres. Nous commençons par une introduction aux conventions du traitement de l'information quantique et un état de l'art des algorithmes de

QPT de QST, et de tomographie de mesures dans le chapitre 1. Puis, dans le chapitre 2, nous introduisons les mesures quantiques que nous réalisons ainsi que nos algorithmes de QST. Nous avons dû établir des algorithmes de QST originaux car aucun des algorithmes de QST de la littérature n'est adapté aux contraintes que nous allons nous donner (états purs et mesures non-intriquées).

Le chapitre 3 est le cœur du manuscrit, il décrit les algorithmes de QPT que nous proposons. Dans la section 3.1, nous commençons par décrire les configurations de QPT que nous considérons (plus générales que la configuration de la figure 1). Nous proposons ensuite une méthode qui (si la QST est effectuée sans erreurs) nous permet d'estimer le processus sans erreurs dans tous les cas où la condition de la section 3.2 est vérifiée. Dans la section 3.2 nous montrons que cette condition est nécessaire et suffisante pour que le processus soit identifiable de façon unique avec les mesures effectuées si la QST est effectuée sans erreur. Cette condition doit être satisfaite pour tout algorithme de QPT, pas seulement pour le nôtre. Si elle n'est pas satisfaite, il existe plusieurs processus distincts qui sont indiscernables avec le dispositif de QPT. En pratique, il y a des erreurs de QST (dues aux imperfections du modèle et au fait que l'on ne fasse qu'un nombre fini de mesures pour estimer les états), et l'algorithme de la section 3.1, en plus d'être très rapide, limite leur impact, à défaut d'être idéal d'un point de vue du maximum de vraisemblance. Si l'on fait confiance au modèle des mesures et que l'on cherche un meilleur estimateur, alors, nous proposons, dans la section 3.3 un algorithme adapté au modèle des erreurs qui identifie tous les paramètres du processus en maximisant la vraisemblance des mesures. La vraisemblance est maximisée par un algorithme de type descente de gradient qui a besoin d'un point initial proche de l'optimum, nous utilisons le résultat de la section 3.1. La méthode de la section 3.3 n'est donc pas autonome.

Dans le chapitre 4 nous levons pour la première fois (dans ce manuscrit) l'hypothèse que les mesures que l'on effectue sont conformes au modèle. Nous proposons un algorithme de tomographie de mesures adapté à nos types de mesures pour estimer les paramètres des mesures que l'on effectue. Nous expliquons aussi comment les algorithmes de QST et QPT peuvent être adaptés si l'on connaît ces paramètres. Finalement dans le chapitre 5, nous testons les algorithmes que nous avons introduits, en simulation et sur des données réelles.

Chapitre 1

Etat de l'art

Sommaire

Nous commençons l'état de l'art par présenter, dans la section 1.1, les enjeux, et formalisme mathématique du traitement de l'information quantique que nous utilisons dans la présente thèse. Puis nous présentons rapidement les travaux de la littérature sur la tomographie d'état et de mesures quantiques, dans les sections 1.2 et 1.3. La tomographie de processus est le cœur du manuscrit, et nous y consacrons toutes les autres sections du présent chapitre. Nous commençons par les différentes familles d'algorithmes de tomographie de processus quelconques de la section 1.4 jusqu'à la section 1.10. Nous avons choisi de nous concentrer sur la tomographie de processus unitaires. Dans la section 1.12, nous justifions ce choix et présentons les algorithmes adaptés à ce problème qui existent dans la littérature.

1.1 Enjeux et conventions

Les ordinateurs quantiques ont été imaginés pour la première fois par P. Benioff en 1980 [Ben80] comme une machine de Turing qui agit sur des états quantiques. Au début, la réalisation d'ordinateurs quantiques n'était motivée que par la nécessité de simuler des systèmes quantiques dont la dimension peut exploser avec le nombre de particules considérées. Ces applications sont toujours très intéressantes, notamment pour la simulation de molécules [OSS+21].

Par ailleurs, un champ entier du traitement de l'information quantique vise à exploiter les propriétés des systèmes quantiques pour résoudre en des temps polynomiaux des problèmes non-polynomiaux pour des ordinateurs classiques. L'exemple le plus connu est l'algorithme de P.W. Shor [Sho94] pour calculer le logarithme discret en un temps polynomial. Si les ordinateurs quantiques parviennent à l'implémenter efficacement sur un grand nombre de qubits, les algorithmes de cryptographie asymétrique seront vulnérables. Il y a d'autres applications, comme l'algorithme de Grover pour la recherche non-structurée dans une base de données [Gro96], la résolution de problèmes d'optimisation complexe avec le recuit simulé quantique [MN08], la résolution de système linéaire de très grande taille [HHL09], et l'amélioration d'algorithmes d'apprentissage [BWP+17].

Le traitement de l'information quantique est aussi utile dans le domaine des télécommunications [Phi23], [IB05]. L'exemple classique est la distribution de clé cryptographique [BB14], qui

commence à se développer dans l'industrie. Elle permet aux utilisateurs de savoir si un tiers a fait des mesures dans le canal pendant la distribution de la clé.

1.1.1 Qubit

Dans un ordinateur quantique l'information est généralement encodée sous forme de bits quantiques ou qubits, l'équivalent des bits dans les ordinateurs classiques. Il existe plusieurs architectures matérielles qui peuvent réaliser des qubits (électrons [VE19], photons [KMN+07], piège à ions [KMW02], supraconducteurs [HWFZ20]). Il existe d'autres blocs de base (qutrits ou qudits d'une autre dimension [WHSK20]) mais ils sont plus rarement utilisés et nous n'avons pas adapté nos algorithmes à ces architectures "niches". L'état d'un qubit évolue dans un espace de Hilbert de dimension 2. Dans le cas où cet état $|\phi\rangle$ est pur, c'est une superposition des états $|0\rangle$ et $|1\rangle$ (qui forment notre base de référence) avec les coefficients complexes c_1 et c_2 tels que $|c_1|^2 + |c_2|^2 = 1$: $|\phi\rangle = c_1 |0\rangle + c_2 |1\rangle$. Les modules au carré de c_1 et c_2 sont les probabilités d'obtenir la valeur associée à $|0\rangle$ ou $|1\rangle$ (souvent 0 ou 1) quand on fait une mesure dans la base de référence ; la base de référence dépend de l'architecture, quand le qubit représente le spin d'un électron par exemple la mesure de référence est la mesure de la composante du spin de l'électron dans une direction donnée (traditionnellement Z). Il est possible de faire d'autres types de mesures que la mesure de référence, mais quelle que soit la mesure que l'on fait, il n'y aura que 2 résultats possibles (les mesures triviales avec 1 résultat possible ne sont pas considérées) que l'on renomme 0 et 1.

1.1.2 Groupe de qubits

Nous considérons aussi des ensembles de plusieurs qubits. Un état pur d'un tel ensemble est une superposition de $d = 2^{n_{qb}}$ (n_{qb} est le nombre de qubits et d est la dimension de l'espace de Hilbert) états $|0...0\rangle, ..., |1...1\rangle$ de la base de référence : $|\phi\rangle = c_1 |0...0\rangle + c_2 |0...01\rangle + ... + c_d |1...1\rangle$. Comme avec 2 qubits, la somme des modules au carré des coefficients vaut 1, et quand on réalise une mesure sur les qubits dans la base de référence, la probabilité de mesurer 0...0 est $|c_1|^2$, la probabilité de mesurer 0...01 est $|c_2|^2$ etc. Dans la suite du présent manuscrit, on utilisera des vecteurs d'état plutôt que des kets ($|.\rangle$), et par abus de langage, on appellera le vecteur $\boldsymbol{v} = \begin{pmatrix} c_1 \\ \vdots \\ c_d \end{pmatrix}$ "l'état" (sous-entendu du système quantique) au lieu de "le vecteur qui représente l'état" .

1.1.3 Mesures projectives

En physique quantique, dans un système de dimension finie, un type de mesure $\mathcal{M}$ à valeur réelle est défini par une matrice hermitienne de taille d : $\mathbf{H}_{\mathcal{M}}$ (à ne pas confondre avec l'hamiltonien du système que nous introduirons en section 1.1.6). Les valeurs propres $\{\lambda_k\}_k$ de $\mathbf{H}_{\mathcal{M}}$ sont les résultats possibles de la mesure, et les vecteurs propres $\{\boldsymbol{e}_k\}_k$ correspondent aux états pour lesquels le résultat de la mesure est connu avec probabilité 1. Quand on réalise une mesure alors que le système est dans l'état $\boldsymbol{e}_k$, on obtient λ_k de façon certaine. Dans le cas général, si on mesure l'état représenté par $\boldsymbol{v}$, la probabilité d'avoir λ_k comme résultat de mesure est la norme au carré de la projection de $\boldsymbol{v}$ sur l'espace propre associé à λ_k. Et l'espérance de la mesure si on mesure $\boldsymbol{v}$ est $\boldsymbol{v}^* \mathbf{H}_{\mathcal{M}} \boldsymbol{v}$. Certains articles [CDJ+13] , [MJZ+16], [CKW+16] n'utilisent que cette espérance. Si $\mathcal{M}$ a strictement plus de 2 résultats possibles (strictement plus de 2 valeurs propres distinctes), cette approche est sous optimale car il y a strictement plus d'information dans les probabilités (théoriques ou empiriques) de chaque résultat que dans l'espérance (théorique ou calculée avec

une moyenne). Par exemple, si un type de mesure est représenté par $\mathbf{H}_{\mathcal{M}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}$ et que l'on mesure l'état $\boldsymbol{v} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, alors, savoir que l'espérance de la mesure est $2,5$ n'est pas aussi informatif que de savoir que les trois résultats de mesures 1, 2, 3, 4 sont équiprobables avec des probabilités de $0,25$. Si $\mathcal{M}$ a 2 résultats possibles, alors l'espérance apporte autant d'information que les probabilités et il est logique de travailler avec elle. Mais les mesures qui n'ont que 2 résultats possibles ne sont pas optimales pour $n_{qb} > 1, d > 2$ comparé à une mesure à d résultats possibles qui va apporter plus d'information sur le système, pour le même coût. En dehors de la bibliographie, nous ne considérerons que des mesures projectives à d résultats possibles.

1.1.4 Mesures projectives à d résultats possibles

Pour un type de mesure à d résultats possibles, la matrice hermitienne $\mathbf{H}_{\mathcal{M}}$ associée a d valeurs propres distinctes et les espaces propres sont des droites. Comme dans le cas général de la section 1.1.3, la probabilité d'obtenir λ_k en mesurant $\boldsymbol{v}$ est la norme au carré de la projection de $\boldsymbol{v}$ sur l'espace propre associé à λ_k. Or, si on a d résultats possibles, le vecteur propre est porté par la droite de vecteur directeur $\boldsymbol{e}_k$, la norme de la projection s'écrit donc : $\|\boldsymbol{e}_k^* \boldsymbol{v}\|^2$.

On définit la matrice $\mathbf{E} = \begin{bmatrix} \boldsymbol{e}_1 & ... & \boldsymbol{e}_d \end{bmatrix}^*$. Avec cette définition, les probabilités d'avoir les résultats λ_1, ..., λ_d sont contenues (dans l'ordre) dans le vecteur $|\mathbf{E}\boldsymbol{v}|^2$ où $|.|^2$ est le module au carré terme à terme. $\mathbf{E}$ est utilisé dans la littérature, mais pas toujours défini très clairement, et toujours appelé différemment ([GCE+15] l'appelle la base des mesures par exemple).

Les résultats possibles λ_1, ..., λ_d peuvent être renommés $\{0, ..0\}$, $\{0, ..01\}$, ..., $\{1, ..1\}$ afin que le résultat d'une mesure réalisée sur un système de qubits soit représenté par une chaîne de bits. Avec cette convention on sort du cadre des mesures à valeurs réelles, mais c'est sans importance, car ce ne sont que les probabilités et les vecteurs propres contenus dans $\mathbf{E}$ qui apportent de l'information quand on effectue une mesure, les valeurs possibles que cette dernière peut prendre peuvent être renommées en n'importe quelles d autres valeurs distinctes sans que l'on perde d'information dans les traitements que nous considérons dans ce manuscrit.

Dans le reste de ce manuscrit, mesurer un vecteur d'état "dans la base $\mathbf{E}$" signifie que l'on mesure le vecteur avec une mesure à d résultats possibles $\{0, ..0\}$, $\{0, ..01\}$, ..., $\{1, ..1\}$ dont les vecteurs propres sont les transconjuguées des lignes de $\mathbf{E}$ (dans l'ordre).

1.1.4.1 Mesures non intriquées

En pratique, toutes les mesures ne sont pas aussi faciles à réaliser. Les plus faciles sont les mesures non intriquées (ou locales), pour les réaliser il faut mesurer chaque qubit indépendamment et concaténer les résultats. Les matrices de vecteurs propres associées sont décomposables en produits tensoriels : $\mathbf{E} = \mathbf{E}_1 \otimes ... \otimes \mathbf{E}_{n_{qb}}$. D'autres mesures, dites globales ou intriquées peuvent être réalisées mais seulement en utilisant des portes quantiques d'intrication. Comme notre objectif est d'identifier les portes sans nous baser sur des portes de références (comme les portes d'intrication), nous nous privons de ces mesures globales.

1.1.4.2 Effets de la mesure

Il n'y a pas vraiment de consensus sur ce qu'un état quantique devient après être mesuré. Les livres d'introduction à la physique quantique ou au traitement de l'information quantique comme [NC00] affirment que, après la mesure, l'état est fixé au vecteur propre associé à la mesure observée, par exemple, si on mesure $0,..0$, l'état est fixé à la transconjuguée de la première ligne de $\mathbf{E}$, et présentent cela comme un postulat de la mécanique quantique. En pratique ce postulat est débattu, [Zur03] voit la mesure comme l'ouverture d'un système fermé qui transforme (einselection) les états mesurés de façon plus complexe que le postulat de [NC00]. Dans [Bal20], L. Ballentine critique assez violemment ce postulat (wavefunction collapse). De façon générale, les algorithmes de traitement de l'information (ce qui comprend les travaux de la présente thèse) ne réutilisent jamais un système après une mesure à cause de l'incertitude sur l'état du système après la mesure. Par ailleurs, il existe des mesures destructrices (sur des photons par exemple) pour lesquelles parler de "l'état du système après la mesure" n'a pas de sens.

1.1.4.3 Mesures répétées

Il existe des cas d'utilisation dans lesquels faire une seule mesure peut suffire, par exemple, l'estimation de phase (voir Section 5.3 de [NC00]). Pour ce problème, on sait que l'état du système n'a que d valeurs possibles et ces valeurs forment une base orthonormée de $\mathbb{C}^d$. Donc si on mesure dans cette base, on saura avec certitude sans avoir à répéter l'expérience dans lequel des d états possibles est le système (car les probabilités de chaque résultat sont contenues dans le produit de $\mathbf{E}$ et l'état qui contient un 1 à l'indice que l'on cherche et $d-1$ zéros ailleurs). Si on se ramène à 1 qubit, c'est comme si on avait l'a priori que $\boldsymbol{v}$ ne peut prendre que les valeurs $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ ou $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$. Dans ce cas, on mesure l'état dans la base $\mathbf{E} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$, si on obtient 0, on sait que $\boldsymbol{v} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$, et si on obtient 1, $\boldsymbol{v} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$.

Mais ce cas est plus l'exception que la règle, en général on n'a pas d'a priori suffisamment fort sur l'état pour qu'une unique mesure dans une base fixée apporte une information conséquente. Dans ce cas, pour que la mesure dans la base $\mathbf{E}$ (associée à un type de mesure $\mathcal{M}$) nous donne une information utile sur l'état, il faut la répéter n_c fois pour avoir une estimée ($\widehat{\boldsymbol{p}^{\mathcal{M}}}$) des probabilités contenues dans le vecteur des probabilités $\boldsymbol{p}^{\mathcal{M}} = |\mathbf{E}\boldsymbol{v}|^2$. Par exemple, avec $\boldsymbol{v} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$, si on effectue $n_c = 100$ mesures dans la base de référence ($\mathbf{E}$ est l'identité), et que l'on obtient 47 fois 0 et 53 fois 1, alors on peut estimer que $|\mathbf{E}\boldsymbol{v}|^2$ vaut $\widehat{\boldsymbol{p}^{\mathcal{M}}} = \begin{pmatrix}0.47\\0.53\end{pmatrix}$, la vraie valeur est $\boldsymbol{p}^{\mathcal{M}} = \begin{pmatrix}0.5\\0.5\end{pmatrix}$.

Le problème évident de cette approche est qu'effectuer une mesure sur un état le modifie d'une façon non maîtrisée (voir section 1.1.4.2). Un état ne peut donc être mesuré qu'une fois avant d'être mis au rebut. Pour pouvoir répéter la mesure, il faut donc répéter l'expérience en travaillant avec n_c copies de l'état à mesurer. Cette approche est très commune avec les algorithmes de traitement de l'information quantique (voir section 1.5.2 de [NC00]).

1.1.5 Intrication

On dit que des qubits sont non-intriqués quand l'état composé de l'ensemble des qubits se décompose en produit tensoriel d'états mono-qubit. Par exemple, pour 2 qubits, l'état $\boldsymbol{v} = \begin{pmatrix}c_1\\c_2\\c_3\\c_4\end{pmatrix} =$

$\frac{1}{2\sqrt{2}}\begin{pmatrix}1\\\sqrt{3}\\1\\\sqrt{3}\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \otimes \frac{1}{2}\begin{pmatrix}1\\\sqrt{3}\end{pmatrix}$ est non intriqué. Pour une paire de qubits non-intriqués, mesurer un des deux qubits (quel que soit le type de mesure) n'apporte aucune information sur la mesure que l'on aurait faite sur l'autre qubit. Dans l'exemple d'état désintriqué que l'on a donné, sachant que l'on a mesuré 0 sur le premier qubit, la probabilité de mesurer 0 sur le deuxième est $\frac{c_1^2}{c_1^2+c_2^2} = 1/4$ (Bayes) et, si on avait mesuré 1 sur le premier, la probabilité serait $\frac{c_3^2}{c_3^2+c_4^2} = 1/4$. Les probabilités sont les mêmes quel que soit le qubit que l'on prend pour référence et le type de mesure dont on calcule les probabilités.

Des qubits qui ont été créés indépendamment dans des systèmes séparés sont forcément non-intriqués, car un des postulats de base de la mécanique quantique est que l'état global du systèmes séparés est le produit tensoriel des états de chaque système. Bien entendu, tous les états ne sont pas non-intriqués, l'exemple classique d'état intriqué est le premier état de Bell $\boldsymbol{\phi} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$, l'intrication de cet état est maximale, dans le sens que, dans la bonne base (en l'occurrence la base de référence), mesurer un qubit donne toute l'information sur la mesure que l'on aurait faite sur l'autre qubit. En effet, mesurer cet état dans la base de référence donne 00 ou 11 avec des probabilités $\frac{1}{2}$, donc les résultats des mesures sont les mêmes sur les deux qubits, et on pourrait se contenter de mesurer le premier qubit et ignorer le deuxième sachant que le résultat de mesure sur le deuxième sera identique.

1.1.5.1 Positive-operator-valued measure

Avant de réaliser une mesure, on peut intriquer un système quantique représenté par des qubits avec de nouveaux qubits, la mesure peut ensuite se faire sur un système de plus grande dimension. Comme la dimension a augmenté, on peut avoir plus de d résultats possibles (d est la dimension du système original). Si on s'autorise ce type de dispositif, on peut effectuer n'importe quel type de mesure caractérisée par un POVM (positive-operator-valued measure), voir la Box 2.5 de [NC00]. Le POVM est paramétré par n_p (un entier quelconque qui peut être supérieur à d) matrices hermitiennes positives $\mathbf{P}_1, ..., \mathbf{P}_{n_p}$ telles que $\sum_{k=1}^{n_p} \mathbf{P}_k = \mathbf{I}_d$. Pour une mesure projective, $n_p = d$ et les $\mathbf{P}_k$ sont les projections sur les espaces propres de $\mathbf{H}_{\mathcal{M}}$. Les probabilités de chaque résultat possible quand on mesure $\boldsymbol{v}$ sont $p_1(\boldsymbol{v}), ..., p_{n_p}(\boldsymbol{v})$, elles ne dépendent que des $\mathbf{P}_k$ et de $\boldsymbol{v}$: $p_k(\boldsymbol{v}) = \boldsymbol{v}^*\mathbf{P}_k\boldsymbol{v}$.

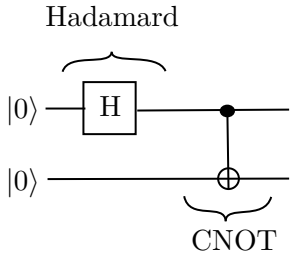
Les POVM sont utilisés dans la littérature, et nous les mentionnerons dans notre revue de la littérature, mais nous ne nous autorisons que des mesures projectives locales dans nos travaux originaux. Ce choix est motivé par le fait que réaliser un POVM non-trivial (i.e. qui n'est pas une mesure projective) en pratique est encore plus complexe que les mesures intriquées que nous avons renoncé à utiliser.

1.1.6 Évolution des systèmes fermés à hamiltonien constant

Dans un système fermé avec un hamiltonien constant, l'équation de Schrödinger $i\hbar\frac{d\boldsymbol{v}}{dt} = \mathbf{H}\boldsymbol{v}$ (où la matrice hermitienne $\mathbf{H}$ est l'hamiltonien du système, i est l'unité imaginaire et $\hbar$ est la constante de Plank réduite) est facile à résoudre formellement (équation différentielle linéaire). La solution est : $\boldsymbol{v}(t) = e^{-\frac{i}{\hbar}\mathbf{H}t}\boldsymbol{v}(0)$ (e est l'exponentielle matricielle). L'évolution de l'état du système est donc linéaire : $\boldsymbol{v}(t+\Delta_t) = \mathbf{M}\boldsymbol{v}(t)$ où $\mathbf{M} = e^{-\frac{i}{\hbar}\mathbf{H}\Delta_t}$ est une matrice unitaire (l'exponentielle complexe de i multiplié par une matrice hermitienne est unitaire) qui ne dépend que de Δ_t

et de l'hamiltonien. En général, c'est ainsi que sont réalisées les portes quantiques. Ces portes appliquent une transformation unitaire à un ou plusieurs qubits, par exemple :

- La porte de Hadamard pour 1 qubit : $\mathbf{M} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$. Cette porte est très utilisée car elle envoie les qubits initialisés à $\begin{pmatrix}1\\0\end{pmatrix}$ (ou $|0\rangle$) sur l'état superposé $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ dont les algorithmes qui exploitent la superposition des états quantiques vont se servir.
- Les portes changement de phase pour 1 qubit : $\mathbf{M} = \begin{pmatrix}1 & 0\\ 0 & e^{i\theta}\end{pmatrix}$. Leurs actions sont invisibles si on mesure dans la base de référence, mais pas dans les autres bases.
- La porte CNOT (controlled NOT) pour 2 qubits : $\mathbf{M} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$. Très utile pour intriquer des qubits, elle envoie une paire de deux qubits desintriqués initialés à $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$ sur le premier état de Bell : $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$. Cette préparation peut être réalisée avec le circuit suivant :



L'ensemble des portes de Clifford est défini comme l'ensemble de ces portes qui combinent les portes CNOT, Hadamard et les portes de phase avec $\theta = \frac{\pi}{2}$.

1.1.7 Applications pour les ordinateurs quantiques

Il n'existe pas encore de processeur quantique qui joue le role de l'Intel 4004 de 1971 pour les ordinateurs classiques. La situation actuelle de l'informatique quantique est comparable à celle des années 1950-1960 pour l'informatique classique, après l'invention des transistors mais avant l'invention des microprocesseurs. Les implémentations matérielles des algorithmes quantiques consistent le plus souvent à initialiser tous les qubits à $|0\rangle$ puis à appliquer des portes pour modifier les valeurs et apporter des intrications. Des "processeurs quantiques" ne peuvent être réalisés physiquement qu'en utilisant ces architectures avec des portes quantiques de petite taille comme bloc de base qui servent à intriquer des qubits voisins [AAB+19] [XCZ+18] [WBC+21] [DDH+22]. Il est rare de voir des groupes de plus de 5-6 qubits intriqués même avec des systèmes de quelques dizaines/centaines de qubits car, physiquement, les qubits sont placés sur des architectures à deux dimensions où seuls les qubits adjacents peuvent être intriqués. Il y a des contre-exemples, notamment [GCZ+19] où les auteurs intriquent 12 qubits avec des portes à 2 qubits, mais l'état final n'est que très peu maîtrisé (fidélité très faible).

La réalisation des portes quantiques unitaires est donc un sujet central pour l'informatique quantique. Estimer les paramètres (éléments de la matrice $\mathbf{M}$) d'une porte qui a été implémentée est très important pour que les portes quantiques réalisées soient conformes, i.e. que le comportement effectif de la porte soit suffisamment proche du comportement désiré. L'objectif de cette thèse est d'évaluer les paramètres d'une porte quantique donnée en mesurant son impact sur des états d'entrée inconnus. Ce choix de considérer des états d'entrée inconnus est motivé par la difficulté à préparer des états prédéterminés de manière précise.

1.1.8 Imperfections du système

Les notions de physique quantique que nous avons introduites sont suffisantes pour comprendre la quasi totalité des contributions originales du présent document. Mais pour comprendre l'état de l'art, il est nécessaire d'introduire la notion d'opérateurs densité et d'opérateurs de Kraus.

1.1.8.1 Opérateur densité pour modéliser des mélanges statistiques dans les systèmes fermés

Comme on l'a vu dans la section 1.1.4.3 il est généralement nécessaire de travailler sur n_c copies d'un état pour pouvoir répéter un type de mesure n_c fois. Or la préparation d'un état, même si l'implémentation matérielle est très précise, n'est jamais exactement répétable et peut être approximée comme un phénomène aléatoire. Il existe une densité $f_{\boldsymbol{v}}$ telle que, quel que soit le sous-ensemble $\mathcal{E}$ de l'espace de Hilbert, la probabilité que l'état préparé $\boldsymbol{v}$ soit dans $\mathcal{E}$ vaut $\int_{\mathcal{E}} f_{\boldsymbol{v}}(\boldsymbol{v})d\boldsymbol{v}$. Dans la littérature, il est plus courant de considérer des densités $f_{\boldsymbol{v}}$ discrètes (section 2.4 de [NC00]) : l'état peut prendre n valeurs discrètes avec des probabilité $p_1, ..., p_n$, c'est un cas particulier d'une densité continue (avec des Diracs). La modélisation que nous allons décrire est valable dans les deux cas (densité continue et discrète).

Soit un type de mesure projective à d résultats possibles $\mathcal{M}$ et une matrice de vecteurs propres $\mathbf{E}_{\mathcal{M}}$ (définie en section 1.1.4) qui lui est associée. On prépare n_c copies de l'état et on fait l'hypothèse que les copies sont des réalisations iid (indépendantes identiquement distribuées) de la variable aléatoire représentée par $f_{\boldsymbol{v}}$. En pratique, quelle que soit $f_{\boldsymbol{v}}$, les n_c résultats de mesure sont iid (car on effectue une opération identique sur des états iid) et ont d résultats possibles. La statistique des mesures est donc entièrement caractérisée par les probabilités théoriques des d résultats possibles (ce sont des lois multinomiales avec d résultats possibles et une seule répétition). Ces probabilités sont les mêmes pour chaque résultat de mesure (car loi iid). Si $\boldsymbol{v}$ était déterministe, ces probabilités vaudraient $|\mathbf{E}_{\mathcal{M}}\boldsymbol{v}|^2$, or, ici, $\boldsymbol{v}$ est aléatoire, les probabilité théoriques que l'on peut estimer avec l'ensemble des n_c résultats de mesures sont donc $\boldsymbol{p}^{\mathcal{M}} = \mathbb{E}_{\boldsymbol{v}}\left(|\mathbf{E}_{\mathcal{M}}\boldsymbol{v}|^2\right)$ où $\mathbb{E}_{\boldsymbol{v}}$ est l'espérance sur le vecteur aléatoire $\boldsymbol{v}$.

Soit $j \in \{1, ..., d\}$, $\boldsymbol{e}_j$ est la j-ième ligne de $\mathbf{E}_{\mathcal{M}}$ et $p_j^{\mathcal{M}}$ est le j-ième élément de $\boldsymbol{p}^{\mathcal{M}}$. On a : $p_j^{\mathcal{M}} = \mathbb{E}_{\boldsymbol{v}}\left(|\boldsymbol{e}_j\boldsymbol{v}|^2\right) = \boldsymbol{e}_j\mathbb{E}_{\boldsymbol{v}}\left(\boldsymbol{v}\boldsymbol{v}^*\right)\boldsymbol{e}_j^*$. Donc quel que soit le type de mesure $\mathcal{M}$, toutes les lignes de $\boldsymbol{p}^{\mathcal{M}}$ (qui caractérisent entièrement la statistique des mesures) ne dépendent que de $\mathbb{E}_{\boldsymbol{v}}(\boldsymbol{v}\boldsymbol{v}^*)$. Donc quelle que soit la densité $f_{\boldsymbol{v}}$ de $\boldsymbol{v}$ (vu ici comme une variable aléatoire), la nature des mesures quantiques fait que la statistique des mesures ne dépend que de $\mathbb{E}_{\boldsymbol{v}}(\boldsymbol{v}\boldsymbol{v}^*)$. Donc toutes les distributions de $\boldsymbol{v}$ pour lesquelles $\mathbb{E}_{\boldsymbol{v}}(\boldsymbol{v}\boldsymbol{v}^*)$ est le même sont indiscernables avec des mesures quantiques, et il est impossible[1] de faire la différence entre les distributions quelles que soient les opérations que l'on fait sur le système pour en extraire de l'information par une mesure.

On appelle $\boldsymbol{\rho} = \mathbb{E}_{\boldsymbol{v}}(\boldsymbol{v}\boldsymbol{v}^*)$ la matrice densité de l'état mélange, comme nous venons de l'expliquer, elle caractérise entièrement l'état. Pour une matrice densité $\boldsymbol{\rho}$ donnée, il existe une infinité de densités de probabilité de $\boldsymbol{v}$ associées. Nous allons décrire la densité la plus simple

[1] tant que l'on fait l'hypothèse que la distribution de $\boldsymbol{v}$ est échantillonée de façon iid pour chaque état préparé.

associée à une matrice densité $\boldsymbol{\rho}$ générique. Soient $\{\boldsymbol{v}_j^{\boldsymbol{\rho}}\}_{j\in[\![1,d]\!]}$ et $\{\lambda_j\}_{j\in[\![1,d]\!]}$ les vecteurs propres et valeurs propres de $\boldsymbol{\rho}$, les $\{\lambda_j\}_{j\in[\![1,d]\!]}$ sont des réels positifs ($\boldsymbol{\rho}$ est hermitienne positive) qui somment à un (la trace de $\boldsymbol{\rho}$ est l'espérance de la norme au carré de $\boldsymbol{v}$ donc 1), on peut donc les interpréter comme des probabilités. Définissons une densité discrète sur les $\{\boldsymbol{v}_j^{\boldsymbol{\rho}}\}_{j\in[\![1,d]\!]}$ pour $\boldsymbol{v}$: $\mathbb{P}(\boldsymbol{v}=\boldsymbol{v}_j^{\boldsymbol{\rho}})=\lambda_j$ ($\mathbb{P}$ est la probabilité). On a bien $\mathbb{E}_{\boldsymbol{v}}(\boldsymbol{v}\boldsymbol{v}^*)=\sum_j(\lambda_j\boldsymbol{v}_j^{\boldsymbol{\rho}}\boldsymbol{v}_j^{\boldsymbol{\rho}*})=\boldsymbol{\rho}$.

On a donc montré (comme [NC00] en section 2.4) que (i) tout état mélange était entièrement caractérisé par sa matrice densité $\boldsymbol{\rho}$ (que l'on appelle aussi "opérateur densité" par abus de langage), et que (ii) tout se passe comme si on avait affaire à un état $\boldsymbol{v}$ aléatoire à valeurs discrètes qui peut prendre pour valeur tous les vecteurs propres de $\boldsymbol{\rho}$ avec comme probabilité les valeurs propres associées.

1.1.8.2 Opérateur densité pour modéliser les systèmes ouverts

Les états mélanges et les opérateurs densité servent aussi pour modéliser les systèmes ouverts (systèmes qui interagissent avec l'extérieur). Dans notre cas, avec des systèmes qui doivent servir pour réaliser des portes quantiques, avoir un système ouvert n'est pas désirable. Mais il est impossible d'isoler parfaitement le système considéré pour qu'il n'y ait aucune interaction avec l'extérieur. Quand ces interactions ne sont pas négligeables, on doit les modéliser. L'univers est un système fermé [DLSS13], donc tous les systèmes ouverts peuvent être vus comme une sous partie d'un système fermé. Par exemple, si on a une paire de qubits intriqués dans le premier état de Bell $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$ parfaitement isolée du reste de l'univers, elle peut être considérée comme un système fermé. Mais si on ne peut observer que le deuxième qubit, c.a.d. en pratique, les mesures que l'on fait sont sur les deux qubits, mais on n'observe que le deuxième bit du résultat (00 est indiscernable de 10 et 01 est indiscernable de 11), alors, ce deuxième qubit constitue un système ouvert qui interagit avec son environnement (en l'occurrence avec le premier qubit). L'état d'un système ouvert ne peut pas (en général) être représenté par un état pur[2], et [NC00] (dans la Section 2.4.3) montre que tout système ouvert peut être représenté par un opérateur densité, même si l'état du système fermé qui englobe le système en question est un mélange statistique. Dans notre exemple avec les 2 qubits dont seul le deuxième est observable, on peut montrer que l'opérateur densité qui représente le système est $\boldsymbol{\rho}=\frac{1}{2}\mathbf{I}_2$.

1.1.8.3 Mesures et évolution des états représentés par des opérateurs densité

Soit un état mélange représenté par $\boldsymbol{\rho}=\sum_{k=1}^{d}p_k\boldsymbol{v}_k\boldsymbol{v}_k^*$, et une mesure projective $\mathcal{M}$ représentée par la matrice hermitienne $\mathbf{H}_{\mathcal{M}}=\sum_{k=1}^{d}\lambda_k\boldsymbol{h}_k\boldsymbol{h}_k^*$ (voir section 1.1.4). Dans un premier temps, on considère que $\mathcal{M}$ est une mesure à $n_{\mathcal{M}}$ résultats possibles distincts. Dans un premier temps, on suppose que $n_{\mathcal{M}}=d$. On rappelle que si l'état était pur et représenté par $\boldsymbol{v}_k$, alors, $\forall\ \lambda_j, j\in\{1,...,d\}$, la probabilité d'obtenir chaque λ_k quand on fait une mesure de type $\mathcal{M}$ est $|\boldsymbol{h}_k^*\boldsymbol{v}_j|^2=\boldsymbol{h}_k^*\boldsymbol{v}_j\boldsymbol{v}_j^*\boldsymbol{h}_k$. Or l'état mélange se comporte comme un mélange statistique qui vaut $\boldsymbol{v}_k$ avec la probabilité p_k. La probabilité d'obtenir λ_k en mesurant $\boldsymbol{\rho}$ est donc :

$$\sum_{j=1}^{d}p_k\boldsymbol{h}_k^*\boldsymbol{v}_j\boldsymbol{v}_j^*\boldsymbol{h}_k=tr\left(\sum_{k=1}^{d}p_k\boldsymbol{h}_k\boldsymbol{h}_k^*\boldsymbol{v}_j\boldsymbol{v}_j^*\right)=tr\left(\boldsymbol{h}_k\boldsymbol{h}_k^*\boldsymbol{\rho}\right). \tag{1.1}$$

[2]Un état pur (par opposition à un état mélange) est représenté par un vecteur d'état et non par un opérateur densité

Dans le cas général où les valeurs propres de $\mathbf{H}_{\mathcal{M}}$ ne sont pas forcément distinctes (i.e. $n_{\mathcal{M}} < d$), les probabilités d'obtenir λ_k sont :

$$p(\mathcal{M}(\boldsymbol{\rho}) = \lambda_k) = tr\left(\mathbf{P}_k \boldsymbol{\rho}\right). \tag{1.2}$$

où $\mathbf{P}_k$ est la matrice de projection sur l'espace propre associé à λ_k : dans le cas non-dégénéré où toutes les valeurs propres sont distinctes, on a $\mathbf{P}_k = \boldsymbol{h}_k \boldsymbol{h}_k^*$.

À partir de ces probabilités, on peut calculer l'espérance de la mesure :

$$\mathbb{E}(\mathcal{M}) = \sum_{k=1}^{n_{\mathcal{M}}} \lambda_k tr\left(\mathbf{P}_k \boldsymbol{\rho}\right) = tr\left(\left(\sum_{k=1}^{n_{\mathcal{M}}} \lambda_k \mathbf{P}_k\right) \boldsymbol{\rho}\right) = tr\left(\mathbf{H}_{\mathcal{M}} \boldsymbol{\rho}\right). \tag{1.3}$$

On remarque que les expressions des probabilités (1.2) et de l'espérance (1.3) sont linéaires en fonction des éléments de la matrice densité $\boldsymbol{\rho}$. Tous ces calculs sont dans [CN97] en Section 2.4.1. Les auteurs montrent aussi que si l'on place l'état représenté par $\boldsymbol{\rho}(t)$ à l'instant t dans un système qui transforme tout état pur $\boldsymbol{v}(t)$ en $\boldsymbol{v}(t + \Delta_t) = \mathbf{M}\boldsymbol{v}(t)$ après Δ_t, alors l'état en sortie est :

$$\boldsymbol{\rho}(t + \Delta_t) = \sum_{k=1}^{d} p_k \mathbf{M}\boldsymbol{v}_k (\mathbf{M}\boldsymbol{v}_k)^* = \mathbf{M}\boldsymbol{\rho}(t)\mathbf{M}^*. \tag{1.4}$$

1.1.8.4 Opérateurs de Kraus pour modéliser l'évolution de systèmes ouverts

On rappelle l'équation de Schrödinger pour les systèmes fermés : $i\hbar \frac{d\boldsymbol{v}}{dt} = \mathbf{H}(t)\boldsymbol{v}(t)$. En général l'hamiltonien n'est pas forcément constant mais on peut montrer (voir [NC00] Section 2.2.2) que, même dans ce cas, l'évolution de système soumis à l'équation de Schrödinger est unitaire :

$\boldsymbol{v}(t) = \mathbf{M}(t)\boldsymbol{v}(0)$, où $\mathbf{M}(t)$ est une matrice unitaire $\mathbf{M}(t) = e^{-\frac{i}{\hbar}\int_{x=0}^{t} \mathbf{H}(x)dx}$. En particulier, l'état $\boldsymbol{v}(t)$ reste pur s'il était pur initialement.
Par opposition, si l'état du système est initialisé dans un état mélange, alors, la Section 2.4.1 de [NC00] montre que l'opérateur densité s'écrit

$$\boldsymbol{\rho}(t) = \mathbb{E}(\boldsymbol{v}(t)\boldsymbol{v}(t)^*) = \mathbf{M}(t)\boldsymbol{\rho}(0)\mathbf{M}(t)^*. \tag{1.5}$$

Le rang et les valeurs propres de $\boldsymbol{\rho}$ sont donc préservés au cours du temps, et les vecteurs propres de $\boldsymbol{\rho}(t)$ sont ceux de $\boldsymbol{\rho}(0)$ (qui représente l'état initial) multipliés à gauche par $\mathbf{M}(t)$.

Pour un système ouvert, l'équation de Schrödinger n'est plus vraie, l'évolution de $\boldsymbol{\rho}$ n'est plus forcément unitaire ((1.5) n'est plus vraie), et l'état du système est représenté par un opérateur densité dont le rang peut changer. On veut étudier comment l'opérateur densité évolue dans un système ouvert :

Soit $t > 0$, on cherche à caractériser le processus invariant non-unitaire $\boldsymbol{\epsilon} : \mathbb{C}^{d\times d} \longrightarrow \mathbb{C}^{d\times d}$ tel que $\boldsymbol{\epsilon}\left(\boldsymbol{\rho}(0)\right) = \boldsymbol{\rho}(t)$. Les seules propriétés que l'on impose à $\boldsymbol{\epsilon}$ sont les suivantes :

1. $\boldsymbol{\epsilon}$ est linéaire, car l'evolution du système fermé est linéaire (unitaire), et la linéarité est préservée sur un sous-système.

2. $\boldsymbol{\epsilon}$ préserve le caractère hermitien positif des matrices ($\boldsymbol{\rho} \in \mathbb{H}_d^+(\mathbb{C}) \Rightarrow \boldsymbol{\epsilon}(\boldsymbol{\rho}) \in \mathbb{H}_d^+(\mathbb{C})$), car les matrices densité en entrée et en sortie sont hermitiennes positives par définition.

3. $\boldsymbol{\epsilon}$ préserve la trace, car les opérateurs densité en entrée et en sortie ont une trace unitaire.

En anglais $\boldsymbol{\epsilon}$ est une "completely positive trace preserving map" (CPTPM). Ce type de fonction a été étudié dans la littérature, les principales contributions ont été apportées dans les années 1970 par Kraus [Kra71] et Choi [Cho75], pour un bon résumé, voir [RSW02].

D'après la Propriété 1, si $\{\boldsymbol{\rho}_k\}_{k\in\{1,...,d^2\}}$ est une base des matrices de $\mathbb{C}^{d\times d}$ sur le corps des complexes, alors, $\boldsymbol{\epsilon}$ est entièrement caractérisé par les valeurs qu'il prend sur cette base. Donc $\boldsymbol{\epsilon}$ est entièrement caractérisé par la matrice $\boldsymbol{\Lambda} \in \mathbb{C}^{d^2\times d^2}$ telle que :

$$\boldsymbol{\epsilon}(\boldsymbol{\rho}_j) = \sum_{k=1}^{d^2} \lambda_{j,k}\boldsymbol{\rho}_k \tag{1.6}$$

où $\lambda_{j,k}$ contient l'élément en ligne j, colonne k de $\boldsymbol{\Lambda}$. Cette paramétrisation (avec $2d^4$ paramètres réels) est simple et très intuitive, mais elle ne prend pas en compte les Propriétés 2 et 3, une matrice $\boldsymbol{\Lambda}$ quelconque sera associée à un processus qui n'a aucune raison de préserver le caractère hermitien positif et la trace.

Les travaux de [Kra71] et [Cho75] ont montré que les Propriétés 1, 2 et 3 (ensemble) sont équivalentes à :

$$\exists \mathbf{K}_1, ..., \mathbf{K}_{n_k} \in \mathbb{C}^{d\times d} \text{ t.q. } \forall \boldsymbol{\rho} \;\; \boldsymbol{\epsilon}(\boldsymbol{\rho}) = \sum_{j=1}^{n_j} \mathbf{K}_j^*\boldsymbol{\rho}\mathbf{K}_j \text{ et } \sum_{j=1}^{n_j} \mathbf{K}_j\mathbf{K}_j^* = \mathbf{I}_d. \tag{1.7}$$

Et si on enlève la dernière condition : $\sum_{j=1}^{n_j} \mathbf{K}_j\mathbf{K}_j^* = \mathbf{I}_d$, on a simplement l'équivalence aux Propriétés 1 et 2 d'après [RSW02]. Les $\mathbf{K}_j$ sont les opérateurs de Kraus, si $n_k = 1$, alors $\mathbf{K}_j$ doit être unitaire pour préserver la trace, cet opérateur de Kraus correspond à une évolution unitaire. Dans le cas général, $n_k \geq 1$, et on peut montrer qu'il est superflu de considérer $n_k \geq d^2$, on a donc $n_k \in \{1, ..., d^2\}$. Définissons une base de l'espace des matrices complexes de taille d : $\{\mathbf{B}_{j,k}\}_{j,k}$ est telle que $\mathbf{B}_{j,k}$ ne contient que des 0 sauf un 1 en ligne j colonne k. On peut donner les exemples suivants de processus non-unitaires :

1. si $\boldsymbol{\epsilon}(\boldsymbol{\rho}) = tr(\boldsymbol{\rho})\mathbf{B}_{1,1}$, alors $\boldsymbol{\epsilon}$ envoie tous les états sur l'état pur $\begin{pmatrix}1 & 0 & ... & 0\end{pmatrix}^T$ ce qui revient à mettre tous les qubits à $|0\rangle$. On peut vérifier que $\boldsymbol{\epsilon}$ est représenté par les d opérateurs de Kraus suivants $\{\mathbf{K}_j = \mathbf{B}_{j,1}\}_{j\in\{1,...,d\}}$.

2. si $\boldsymbol{\epsilon}(\boldsymbol{\rho}) = \frac{tr(\rho)}{d}\mathbf{I}_d$, alors $\boldsymbol{\epsilon}$ envoie tous les états sur l'état mélange représenté par $\frac{1}{d}\mathbf{I}_d$ qui est l'état "le plus mélangé", il est trés éloigné de tous les états purs. Cet état est un mélange équiprobable de tous les éléments d'une base orthonormale. Si on le mesure avec n'importe quelle mesure projective à d résultats possibles, tous les résultats sont équiprobables, et quelle que soit la transformation unitaire qu'on lui applique, il ne changera pas. On ne peut donc rien tirer de cet état avec des traitements classiques (transformation unitaire et mesures). Ce processus $(\boldsymbol{\epsilon})$ peut représenter une forme extrême de décohérence. La décohérence survient dans les systèmes ouverts au bout d'un certain temps, elle dégrade l'intrication et la pureté des états. On peut vérifier que $\boldsymbol{\epsilon}$ est représenté par les d^2 opérateurs de Kraus suivants $\left\{\mathbf{K}_{j+d(k-1)} = \frac{1}{d}\mathbf{B}_{j,k}\right\}_{j,k\in\{1,...,d\}}$.

Il convient de noter que les opérateurs de Kraus qui représentent une CPTPM $\boldsymbol{\epsilon}$ donnée sont loin d'être uniques et on ne peut pas les rendre uniques en rajoutant des contraintes simples [RSW02]. Par exemple, les opérateurs $\left\{\mathbf{K}_{j+d(k-1)} = \frac{1}{d}\mathbf{B}_{j,k}\right\}_{j,k\in\{1,...,d\}}$ que nous avons donnés dans l'exemple 2 ci-dessus pour $\boldsymbol{\epsilon}(\boldsymbol{\rho}) = \frac{tr(\rho)}{d}\mathbf{I}_d$ peuvent être modifiés en multipliant tous les $\mathbf{K}_j$ par la même matrice unitaire à droite et $\boldsymbol{\rho}$ sera le même. Avec le modèle du processus $(\boldsymbol{\epsilon})$ unitaire, la matrice unitaire $(\mathbf{M})$ qui caractérise le processus $(\boldsymbol{\epsilon}(\boldsymbol{\rho}) = \mathbf{M}\boldsymbol{\rho}\mathbf{M}^*)$ est unique à une phase près.

Dans la littérature, il est populaire de décomposer les opérateurs de Kraus dans une base orthonormée (pour le produit scalaire de Hilbert-Schmidt sur le corps des complexes) des matrices

$\mathbb{C}^{d\times d}$ que l'on appelle $\{\widetilde{\mathbf{K}}_j\}_{j=1,\dots,d^2}$: $\mathbf{K}_j = \sum_{k=1}^{d^2} a_{j,k}\widetilde{\mathbf{K}}_k$. Ainsi, on obtient une représentation équivalente avec une matrice de processus $\boldsymbol{\chi} \in \mathbb{C}^{d^2\times d^2}$ hermitienne positive :

$$\forall \boldsymbol{\rho} \quad \boldsymbol{\epsilon}(\boldsymbol{\rho}) = \sum_{j=1}^{d^2}\sum_{k=1}^{d^2} \chi_{j,k}\mathbf{K}_j^*\boldsymbol{\rho}\mathbf{K}_k \tag{1.8}$$

où $\chi_{j,k} = \sum_{\ell=1}^{d^2} a_{\ell,j}a_{\ell,k}^*$ est l'élément en position (j,k) de $\boldsymbol{\chi}$. $\boldsymbol{\chi}$ est la matrice de processus. Par défaut, dans la suite de la thèse, quand on parle de matrice de processus sans définir la base $\{\widetilde{\mathbf{K}}_j\}_{j=1,\dots,d^2}$, c'est que $\{\widetilde{\mathbf{K}}_j\}_{j=1,\dots,d^2} = \{\mathbf{B}_{k,\ell}\}_{k,\ell=1,\dots,d}$ est la base canonique des matrices (un 1 en position k,ℓ et que des zéros ailleurs avec k,ℓ qui varient et $j = d(\ell-1)+k$).

La matrice de processus a d^4 paramètres réels (matrice hermitienne de taille d^2), c'est plus que le nombre de paramètres de Kraus. On pourrait croire que cette représentation est moins unique que celle qui utilise tous les paramètres de Kraus, mais ce n'est pas le cas. Pour une base donnée, la matrice de processus caractérise le processus de façon unique (voir, par exemple équation (3.2) de [CN97]). Cependant, la matrice a trop de degrés de liberté, et une matrice de processus hermitienne quelconque est associée à un processus linéaire, qui n'est positif que si $\boldsymbol{\chi}$ est positif, et qui n'a aucune raison de préserver la trace. On dit que le processus n'est pas forcément "physique".

1.1.9 Fidelité

La métrique classique pour estimer la proximité entre deux états quantiques représentés par $\boldsymbol{\rho}$ et $\widehat{\boldsymbol{\rho}}$ (matrices hermitiennes positive de trace 1) dans la littérature est la fidélité :

$$f(\boldsymbol{\rho},\widehat{\boldsymbol{\rho}}) = tr\left(\sqrt{\sqrt{\boldsymbol{\rho}}\widehat{\boldsymbol{\rho}}\sqrt{\boldsymbol{\rho}}}\right). \tag{1.9}$$

La racine carrée d'une matrice $\boldsymbol{\rho}$ est définie comme la matrice symétrique qui a les mêmes espaces propres et dont les valeurs propres sont les racines carrées de celles de $\boldsymbol{\rho}$. La fidélité a les propriétés suivantes :

- Elle est symétrique $f(\boldsymbol{\rho},\widehat{\boldsymbol{\rho}}) = f(\widehat{\boldsymbol{\rho}},\boldsymbol{\rho})$.
- Elle est comprise entre 0 et 1, et la valeur maximale (1) est atteinte si et seulement si $\boldsymbol{\rho} = \widehat{\boldsymbol{\rho}}$.
- Pour les états purs : $\boldsymbol{\rho} = \boldsymbol{v}\boldsymbol{v}^*$, $\widehat{\boldsymbol{\rho}} = \widehat{\boldsymbol{v}}\widehat{\boldsymbol{v}}^*$, la fidélité est le carré du module du produit scalaire : $f(\boldsymbol{\rho},\widehat{\boldsymbol{\rho}}) = |\widehat{\boldsymbol{v}}^*\boldsymbol{v}|^2$.

La fidélité peut aussi être utilisée pour calculer la proximité entre deux processus quantiques grâce à l'isomorphisme de Choi–Jamiołkowski qui associe un état mélange $\boldsymbol{\rho}_{\boldsymbol{\epsilon}}$ de taille $2d\times 2d$ (dans un espace de Hilbert de dimension deux fois plus grande que celui dans lequel $\boldsymbol{\epsilon}$ opère). Pour une description de l'isomorphisme de Choi–Jamiołkowski, voir la section 1.5.1. La fidélité d'un processus $\widehat{\boldsymbol{\epsilon}}$ vu comme un estimateur de $\boldsymbol{\epsilon}$ est définie comme : $f(\boldsymbol{\epsilon},\widehat{\boldsymbol{\epsilon}}) = f(\boldsymbol{\rho}_{\boldsymbol{\epsilon}},\boldsymbol{\rho}_{\widehat{\boldsymbol{\epsilon}}})$

1.2 Tomographie d'état

Un bloc de base pour la tomographie de processus est la tomographie d'état. La tomographie d'état a pour objectif d'identifier un état à partir de mesures effectuées sur des copies de cet état. En général, la plupart des algorithmes de tomographie de processus (cela inclut nos contributions) se basent sur la tomographie d'état, au moins pour estimer les états qui sortent de la porte que l'on veut identifier. L'état que l'on veut estimer peut être considéré comme un état pur ou un état mélange.

1.2.1 Tomographie d'état mélange

La plupart des travaux effectués sur la tomographie d'état portent sur les états mélange. La version de base est détaillée dans [NC00] au début de la section 8.4.2. Elle utilise des mesures définies par les opérateurs de Pauli, souvent appelées mesures de Pauli [KKD15], [SRA+13], [GLF+10], [MJZ+16], [CKW+16] et [Wan13]. Cette version est simple et très robuste, mais elle a deux problèmes : le premier est que la matrice densité qu'elle produit est forcément hermitienne, mais peut ne pas être positive, et, en fonction de la façon dont les mesures sont réalisées, la trace peut ne pas être unitaire, on dit alors qu'elle n'est pas physique. Dans [JKMW01] [GKKT20], les auteurs proposent une méthode pour trouver une matrice densité physique (maximum de vraisemblance pour [JKMW01], projection sur l'ensemble des matrices physiques pour [GKKT20]). Le deuxième problème est que l'estimation initiale de la matrice densité nécessite de calculer les moyennes de d^2-1 différents types de mesures de Pauli multi-qubit. Pour chacun des d^2-1 types de mesures, il faut préparer de nombreuses copies de l'état, cela devient très rapidement prohibitif quand le nombre de qubits augmente. Cependant, étant donné que l'état que l'on cherche à estimer est représenté par une matrice hermitienne de taille $d = 2^{n_{qb}}$ et de trace unitaire qui a donc d^2-1 paramètres réels, on ne peut pas espérer faire beaucoup mieux avec des mesures de Pauli multi-qubit. En effet, ces mesures de Pauli n'ont que 2 résultats possibles, on n'évalue donc que 2 probabilités empiriques en répétant une mesure de Pauli, ces deux probabilités somment à 1, il n'y a donc qu'un seul degré de liberté. Si on veut estimer d^2-1 paramètres réels avec seulement des mesures à 2 résultats possibles (ce qui est sous optimal), on ne peut donc pas espérer y arriver avec moins de d^2-1 types de mesures.

Afin d'effectuer la tomographie d'état avec moins de types de mesures, on peut se concentrer sur un sous-ensemble de tous les états mélanges. Depuis les années 2010, les approches parcimonieuses ont été appliquées à la tomographie d'état, et l'hypothèse la plus courante est que la matrice densité $\boldsymbol{\rho}$ représentant l'état a un rang faible. [GLF+10] a introduit une approche parcimonieuse qui nécessite les moyennes de $O(rd\log^2(d))$ types de mesures (pour des mesures à deux résultats possibles), où r est le rang de $\boldsymbol{\rho}$. Quand r est petit (par rapport à d qui correspond à une matrice $\boldsymbol{\rho}$ de rang plein), le fait de n'utiliser que $O(rd\log^2(d))$ types de mesures représente un gain important comparé aux d^2-1 types de mesures de [NC00]. Par la suite, [SRA+13], [KKD15], ont raffiné cette idée. Plus récemment, dans [BDK16], la tomographie d'état à rang borné a été introduite. Elle suppose que le rang r est connu et permet la reconstruction explicite de la matrice densité en utilisant des mesures prédéterminées.

D'autres approches ne font pas d'hypothèses sur $\boldsymbol{\rho}$. En 2014, la SGQT (self guided quantum tomography) a été introduite [Fer14] et développée dans [CFP16], [AFS22]. La SGQT marche quel que soit le rang de $\boldsymbol{\rho}$ et le nombre de types de mesures nécessaire varie de façon très raisonnable avec le nombre de qubits. Le cœur de la méthode est un algorithme d'optimisation stochastique qui choisit à chaque itération le type de mesure (possiblement intriqué) qui sera effectué à la mesure suivante et, si tout se passe bien, la dernière mesure sera la projection sur l'état à identifier. Avec les ordinateurs quantiques actuels, la SGQT est irréaliste (à part avec un seul qubit) car on ne peut pas demander à l'opérateur de faire n'importe quel type de mesure au fur et à mesure que l'expérience progresse. Si ces mesures que l'on doit réaliser sont intriquées (en général c'est le cas) elles ne peuvent être réalisées qu'avec des portes d'intrication, étant donné que nous cherchons à identifier des portes, nous n'utilisons pas ce type de mesure.

1.2.2 Tomographie d'état pur

1.2.2.1 Récupération de phase

Si on fait l'hypothèse que l'état du système est pur, le problème de tomographie d'état change : on cherche le vecteur complexe $\boldsymbol{v}$ tel que les fréquences d'occurrence observées de chaque résultat

possible $\widehat{\boldsymbol{p}}$ (de taille dn_t où n_t est le nombre de types de mesures que l'on réalise) soient le plus proche possible des fréquences d'occurrence théoriques $\boldsymbol{p} = |\mathbf{A}\boldsymbol{v}|^2$ avec $\mathbf{A}$ la concaténation des matrices définissant les bases des différents types de mesures : $\mathbf{A} = \begin{bmatrix} \mathbf{E}_{\mathcal{M}_1} \\ \vdots \\ \mathbf{E}_{\mathcal{M}_{n_t}} \end{bmatrix}$. Fondamentalement, retrouver $\boldsymbol{v}$ à partir de $|\mathbf{A}\boldsymbol{v}|^2$ est un problème de récupération de phase (phase retrieval). Souvent dans la littérature sur la récupération de phase, $|\mathbf{A}\boldsymbol{v}|$ est utilisé à la place de $|\mathbf{A}\boldsymbol{v}|^2$, mais il s'agit du même problème (à moins de considérer du bruit, et le problème est assez complexe sans bruit). Ce problème est connu comme étant compliqué, mais a été étudié extensivement dans la littérature, car il existe d'autres applications (imagerie par rayons X [MISE08] ou par diffraction [BDP+07]). Avant de se poser le problème de la récupération de la phase, il faut se demander si la mesure est injective (sous-entendu à une phase globale près), c'est-à-dire si (pour des types de mesure et un $\mathbf{A}$ donnés) tout vecteur complexe $\boldsymbol{v}$ peut être retrouvé à une phase globale près à partir $|\mathbf{A}\boldsymbol{v}|^2$. Cette question à elle seule est très complexe et il n'existe aucune condition nécessaire et suffisante simple à vérifier sur $\mathbf{A}$ qui garantit l'injectivité. Les travaux de Heinosaari et al. [HMW13] donnent une condition nécessaire pour que l'injectivité soit possible : il faut que le nombre de lignes de $\mathbf{A}$ (qui pour nous est le nombre de fréquences empiriques mesurées que l'on appelle $n_{prob} = n_t d$) soit strictement supérieur à une borne suivante (avec nos notations) : $n_{prob} > 4d - 3 - c(d)n_{qb}$ où $c(d)$ vaut soit 1 soit 2. Bien entendu, ce n'est pas une condition suffisante, et on ne peut pas avoir de conditions suffisantes seulement en regardant le nombre de lignes de $\mathbf{A}$ (il faut au moins que $\mathbf{A}$ soit de rang plein). [BCE06] est assez proche d'une condition suffisante cependant. Il montre que, quel que soit $n_{prob} > 4d - 2$, il existe un ensemble $\mathcal{F} \in \mathbb{C}^{n_{prob} \times d}$ de mesure nulle, tel que si $\mathbf{A} \notin \mathcal{F}$, alors, la mesure est injective. En clair, si $n_{prob} > 4d - 2$, et, si on choisit un $\mathbf{A}$ aléatoire, alors, la probabilité de choisir un $\mathbf{A}$ qui rend la mesure non-injective est nulle. [BCMN14] explique pourquoi il est naturel de penser que $n_{prob} > 4d - 4$ donne la même propriété (sans pouvoir le prouver).

Au delà du problème de l'injectivité des mesures, retrouver $\boldsymbol{v}$ à partir de $|\mathbf{A}\boldsymbol{v}|^2$ (ou $|\mathbf{A}\boldsymbol{v}|$) est un problème connu comme étant complexe. [WdM13] est un très bon article qui présente plusieurs méthodes pour résoudre ce problème. Ces méthodes sont générales, et sont censées fonctionner pour n'importe quelle matrice $\mathbf{A}$.

1.2.2.2 Travaux de Finkelstein

Les articles de la littérature sur la tomographie d'état pur sont assez rares comparés aux articles sur la tomographie d'état mélange. Dans [Fin04], Finkelstein décrit un système capable de distinguer presque tous les états purs avec une matrice $\mathbf{A}$ qui ne contient que $2d$ lignes, c'est moins que la borne de [HMW13], ce qui est normal, car les mesures ne sont pas strictement injectives, mais elles sont injectives sur un ensemble d'état de mesure pleine, on dit qu'il y a un ensemble d'échecs de mesure nulle (zero-measure failure set). Finkelstein propose un algorithme original et adapté à ses mesures pour estimer l'état que l'on cherche. Aucun système proposé dans la littérature pour réaliser la QPT d'état pur n'a moins de lignes pour $\mathbf{A}$ que celui de Finkelstein, mais, en pratique, ce dernier a deux inconvénients de taille :

- La matrice $\mathbf{A}$ de Finkelstein ne peut pas être décomposée en une concaténation verticale de matrices unitaires. Cela veut dire que, en pratique, les $2d$ types de mesures ne peuvent pas être effectués indépendamment avec 2 mesures quantiques à d résultats possibles. On dit que le système de Finkelstein n'exploite pas le pouvoir du parallélisme. En pratique, quelle que soit $\mathbf{A}$, il est toujours possible de trouver des mesures quantiques (à d résultats possibles ou pas) qui permettent d'estimer $|\mathbf{A}\boldsymbol{v}|^2$, on peut par exemple, effectuer autant

de mesures que $\mathbf{A}$ a de lignes (n_{prob}) en mesurant dans les bases $\begin{bmatrix} \boldsymbol{a}_1 \\ \mathbf{R}_1 \end{bmatrix}, ..., \begin{bmatrix} \boldsymbol{a}_{n_{prob}} \\ \mathbf{R}_{n_{prob}} \end{bmatrix}$ (où $\boldsymbol{a}_k$ est la k-ième ligne de $\mathbf{A}$ normalisée et $\mathbf{R}_k$ est une matrice qui complète $\boldsymbol{a}_k$ de façon à ce que la concaténation soit une matrice unitaire). Mais ce n'est pas du tout optimal car on fait n_{prob} mesures qui ont chacune d résultats possibles et on ne retient que le premier résultat de chaque mesure.

- Les d dernières lignes de la matrice $\mathbf{A}$ de Finkelstein sont des vecteurs intriqués, elles correspondent à des mesures intriquées.

1.2.2.3 Travaux de Goychene et al.

Plus récemment Goychene et al. [GCE+15] ont défini un système de tomographie d'état pur. La matrice $\mathbf{A}$ de Goyeneche a $n_{prob} = 4d$ lignes qui sont la concaténation de 4 bases orthonormales. En réalité $n_{prob} = 5d$ mais on peut simplement utiliser les mesures dans les 4 bases définies dans l'équation (2) de [GCE+15], la cinquième base ne sert qu'à vérifier que l'état est bien pur. Le fait que la matrice $\mathbf{A}$ de Goyeneche puisse être décomposée en la concaténation de 4 matrices orthonormales est très positif, cela signifie que $|\mathbf{A}\boldsymbol{v}|^2$ peut être estimée de façon optimale avec 4 types de mesures à d résultats possibles. Cependant ces mesures seront intriquées, les deux premières bases de l'équation (2) de [GCE+15] sont non-intriquées mais pas les deux dernières. Les auteurs reconnaissent que c'est un problème, et remarquent que les deux derniers types de mesures (intriquées) peuvent être réalisés en mesurant, avec les deux premiers types de mesures (non-intriqués), l'état que l'on veut mesurer sur lequel a été appliquée la transformée de Fourier quantique 2 fois. Cet argument apporte de la crédibilité au système de [GCE+15] car la transformée de Fourier quantique est une brique de base (réalisée avec une porte quantique unitaire) du traitement de l'information quantique très étudiée. Mais les réalisations matérielles de cette porte vont forcément être imparfaites et ajouter des erreurs à l'estimée de l'état. C'est le problème qu'ont toutes les mesures intriquées, elles peuvent être réalisées en faisant des mesures non-intriquées sur des états que l'on rend intriqués par une porte quantique, mais fabriquer une bonne porte quantique est censé être une tâche plus complexe à réaliser que la tomographie d'état, c'est donc problématique d'avoir des systèmes de tomographie d'état qui se basent sur l'existence de ces portes. Comme pour Finkelstein, les mesures proposées par Goychene et al. ne sont pas strictement injectives, mais elles ont un ensemble d'échecs de mesure nulle.

Comme Finkelstein, Goychene et al. proposent un algorithme adapté à leurs types de mesures pour estimer l'état. De manière générale, nous n'avons trouvé aucun article de tomographie d'état pur qui utilise les algorithmes génériques de [WdM13], certains ([GCE+15] [CDJ+13] [MJZ+16]) citent [HMW13] pour justifier le nombre de types de mesures choisi, mais les algorithmes de récupération de l'état sont spécifiques aux types de mesures choisis.

1.2.2.4 Autres travaux

Dans [FSC05] Flammia et al. présentent un POVM à $2d$ résultats possibles qui permet d'identifier un état pur. Cette méthode est très populaire mais nous ne la mentionnons qu'ici car elle utilise un POVM et nous préférons les mesures projectives (voir section 1.1.5.1).

Dans [CPF+10], les auteurs considèrent la tomographie d'états de grande dimension qui ont une structure réaliste. Pour des systèmes de plusieurs dizaines ou centaines de qubits, notre modèle avec des vecteurs d'états de taille d a beaucoup de paramètres inutiles. Sur une centaine de qubits, si le premier et le dernier sont physiquement à l'opposé de la puce sur laquelle on travaille, ils ne seront pas intriqués. Il en va de même pour la plupart des qubits (on a rarement plus de 5 ou 6 qubits intriqués). Un meilleur modèle pour représenter ces systèmes

est appelé matrix state product (MPS). Et [CPF+10] propose des algorithmes adaptés à ce modèle. Dans cette thèse, nous nous intéressons à des petits systèmes pour identifier des petites portes, nous n'utiliserons donc pas le modèle MPS ou les algorithmes associés. Dans le même esprit [CW20] propose des méthodes pour obtenir des informations (comme les valeurs de chaque qubit considéré séparément et les niveaux d'intrication entre les différentes paires de qubits) sur des systèmes quantiques de très grande dimension.

La méthode de Goychene et al. semble être la plus reconnue dans la littérature parmi les méthodes classiques de tomographie d'état pur. [BDK16], par exemple, se base sur les 4 types de mesures de Goychene et al. et étend la méthode pour les opérateurs densité de rang borné. D'autres travaux comme [CW20] et [XNK+20] parlent de tomographie d'état pur mais les applications et les méthodes sont très différentes.

[XNK+20] propose un système comparable à [Fer14] où les types de mesures que l'on fait à chaque itération dépendent des résultats des mesures de l'étape précédente.

[CDJ+13] et [MJZ+16] font aussi de la tomographie d'état pur, mais les mesures considérées n'ont pas d résultats possibles. Nous expliquerons en détail dans la partie 2.1.2 pourquoi nous pensons que ces méthodes sont sous-optimales.

1.3 Tomographie de mesure quantique

Tous les algorithmes de tomographie d'état et de processus calculent les paramètres avec des résultats de mesures quantiques. Implicitement ou explicitement, ils font l'hypothèse que les mesures qu'ils réalisent sont conformes au modèle. Des algorithmes de QMT ("quantum measurement tomography" aussi appelés QDT pour "quantum detector tomography") ont donc été établis pour estimer les paramètres de mesures projectives [LSS99] et de POVMs [Fiu01] [LFCR+09]. Tous ces algorithmes estiment des mesures en les réalisant sur des états connus, c'est un problème de régression qui vise à inverser l'équation des probabilités des POVM $p_k(\boldsymbol{v}) = \boldsymbol{v}^*\mathbf{P}_k\boldsymbol{v}$ pour les état purs ou $p_k(\boldsymbol{\rho}) = tr(\mathbf{P}_k\boldsymbol{\rho})$ pour les états mélange, où p_k est la probabilité du k-ième résultat possible, $\boldsymbol{v}$ ou $\boldsymbol{\rho}$ représente l'état (pur ou mélange, en pratique on mesure plusieurs états donc ils prendront plusieurs valeurs) et les matrices $\mathbf{P}_k$ caractérisent le type de mesure à identifier. Le problème est assez similaire à la tomographie d'état, on estime une mesure inconnue à partir d'un état connu à la place d'estimer un état inconnu à partir d'une mesure connue. Le fait que ces deux problèmes soient interdépendants est problématique : les algorithmes de QST s'appuient sur de bonnes mesures pour lesquelles on a besoin de faire de la QMT qui s'appuie sur des bons états, or, on ne peut vérifier que des états sont bons qu'avec la QST. Ce problème est soulevé par les partisans de la gate set tomographie (GST), voir la section 1.8. Pour y remédier, ces derniers proposent d'estimer les états et les mesures simultanément comme nous le verrons en section 1.8. Ce n'est possible qu'à des indéterminations près. L'idée d'estimer les états et les mesures en même temps est reprise par Keith et al. dans [KBGK18], les auteurs se soucient davantage de l'implémentation physique que les défenseurs de la GST. Plutôt que de considérer qu'aucune mesure et aucune préparation d'état n'est assez précise pour être digne de confiance (ce qui est fait dans la GST), Keith at al. choisissent de faire confiance à une initialisation de l'état et à des portes (mono-qubit dans leurs exemples) qui leur servent à créer d'autres états, cela leur permet de faire moins de mesures et de ne pas avoir d'indéterminations. Cette méthode, ainsi que la GST est testée expérimentalement dans [CFYW19].

1.4 Tomographie de processus standard

[PCZ97] et [CN97] ont introduit ce qui a été ensuite appelé la tomographie de processus standard (standard quantum process tomography). L'idée de base est d'estimer les paramètres d'un

processus quantique non-unitaire en appliquant ce processus à des états d'entrée de référence connus, puis d'estimer les états de sorties (avec de la tomographie d'état mélange basique du type [NC00]), et enfin, de faire une régression pour trouver la matrice de processus.

On rappelle que tout processus quantique est linéaire, donc entièrement caractérisé par son action sur une base de $\mathbb{C}^{d\times d}$. Chuang et Nielsen (les auteurs de [CN97]) montrent que, pour tout processus $\boldsymbol{\epsilon}$, on peut estimer tous les $\{\boldsymbol{\epsilon}(\mathbf{B}_{j,k})\}_{j,k\in\{1,...,d\}}$ (où $\mathbf{B}_{j,k}$ est définie dans le glossaire comme la base canonique des matrices) à partir de $\{\boldsymbol{\epsilon}(\boldsymbol{\rho}_\ell^{in})\}_{\ell\in\{1,...,d^2\}}$ pour un ensemble de d^2 d'entrée (purs) $\{\boldsymbol{\rho}_\ell^{in}\}_{\ell\in\{1,...,d^2\}}$ décrit dans [CN97]. Or les $\{\mathbf{B}_{j,k}\}_{j,k\in\{1,...,d\}}$ forment une base. Si on est capable de préparer les états $\{\boldsymbol{\rho}_\ell^{in}\}_{\ell\in\{1,...,d^2\}}$, de leur appliquer $\boldsymbol{\epsilon}$, et d'estimer les états en sortie $\{\boldsymbol{\epsilon}(\boldsymbol{\rho}_\ell^{in})\}_{\ell\in\{1,...,d^2\}}$, on peut donc calculer $\{\boldsymbol{\epsilon}(\mathbf{B}_{j,k})\}_{j,k\in\{1,...,d\}}$, et on a les valeurs de $\boldsymbol{\epsilon}$ sur une base. Ces valeurs permettent de calculer la matrice Λ de (1.6) pour la base des $\{\mathbf{B}_{j,k}\}_{j,k\in\{1,...,d\}}$.

Avec leur équation (3.8) Chuang et Nielsen donnent une méthode pour exprimer la matrice de processus $\boldsymbol{\chi}$ de (1.8) à partir de la matrice Λ de (1.6). L'idée de [PCZ97] est comparable, mais la méthode n'est définie que pour 2 qubits.

Ces méthodes ont deux inconvénients majeurs :

- Elles partent du principe qu'il est possible de créer des états de référence sans erreurs. En pratique, les états de référence sont créés avec des portes quantiques de référence. Étant donné que l'objectif est d'identifier les portes quantiques, le fait d'avoir besoin de bien connaître une porte de référence est problématique.

- Le fait que l'on ne fasse pas d'hypothèse sur $\boldsymbol{\epsilon}$ rend l'algorithme très général (et c'est un point fort) mais le nombre de paramètres à estimer devient facilement prohibitif. Pour $n_{qb} = 2$, la matrice de processus a $d^4 = (2^{n_{qb}})^4 = 256$ paramètres, pour 3 qubits, on passerait à 4096 paramètres.

Un autre problème est que l'estimée de la matrice de processus que l'on obtient peut ne pas être "physique" i.e. le processus associé peut ne pas préserver la positivité ou la trace. On peut projeter dans l'ensemble des processus physiques [SSKKG22], ou alors chercher le processus physique qui maximise la vraisemblance [JFH03].

1.5 Tomographie de processus avec qubits ancillaires

1.5.1 Exploitation directe de l'isomorphisme de Choi-Jamiołkowsky

La tomographie de processus avec qubits ancillaires (Ancilla Assisted Process Tomography (AAPT) en anglais), [DP01] [ABJ+03] [SM14] permet d'estimer les paramètres d'un processus non-unitaire avec le circuit suivant :

$2n_{qb}$ qubits sont initialisées à $|0\rangle$, des états de Bell sont créés avec des portes de Hadamard et des portes CNOT, puis le processus à identifier ϵ est appliqué aux n_{qb} derniers qubits. L'état du système avant que ϵ soit appliqué aux n_{qb} derniers qubits est le suivant (écrit avec des kets et pas avec un vecteur d'état par souci de lisibilité) :

$$|\psi_a\rangle = \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |s_j s_j\rangle \tag{1.10}$$

où s_j est la j-ième chaîne de bit de taille n_{qb} ($s_1 = 0...0, s_1 = 0...01, ..., s_d = 1...1$). En clair, il y a d possibilités équiprobables $(0, ..0, ..., 1...1)$ pour les valeurs que peut prendre la mesure dans la base de référence sur n_{qb} premiers et n_{qb} derniers qubits, mais quand on effectue la mesure dans la base de référence sur les $2n_{qb}$ qubits, les deux valeurs des sous chaînes (formées de n_{qb}

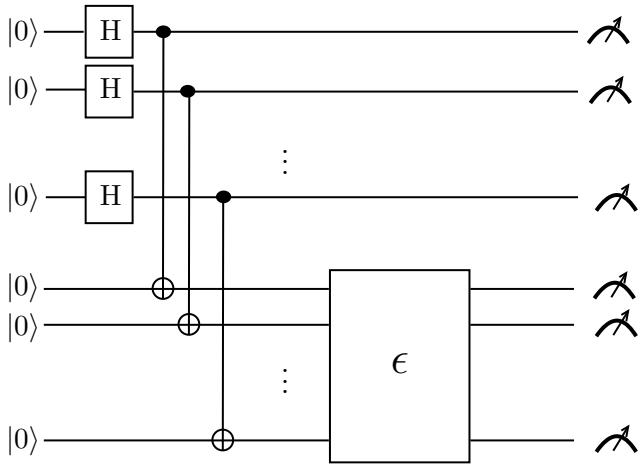


FIGURE 1.1 : Circuit de l'AAPT

caractères) au début et à la fin seront forcément identiques. Pour $n_{qb} = 2$ par exemple, on peut obtenir 0000, 0101, 1010 et 1111 de façon équiprobable.

On peut aussi écrire $\boldsymbol{v}_a$, le vecteur d'état associé à $|\boldsymbol{\psi}_a\rangle$:

$$\boldsymbol{v}_a = \frac{1}{\sqrt{d}} \sum_{j=1}^{d} \boldsymbol{\delta}_j \otimes \boldsymbol{\delta}_j \tag{1.11}$$

où $\boldsymbol{\delta}_j$ est le vecteur de taille d qui vaut 1 en j et 0 ailleurs. Les equations (1.11) et (1.10) correspondent au même état car le vecteur $\boldsymbol{\delta}_j \otimes \boldsymbol{\delta}_j$ correspond au ket $|s_j s_j\rangle$.

Comme ϵ va "ouvrir le système", on doit considérer l'opérateur densité associé à l'état pur de (1.11) :

$$\boldsymbol{\rho} = \frac{1}{\sqrt{d}} \sum_{j=1}^{d} \boldsymbol{\delta}_j \otimes \boldsymbol{\delta}_j \left(\frac{1}{\sqrt{d}} \sum_{j=1}^{d} \boldsymbol{\delta}_j \otimes \boldsymbol{\delta}_j \right)^* = \frac{1}{d} \sum_{j,k=1}^{d} \mathbf{B}_{j,k} \otimes \mathbf{B}_{j,k} \tag{1.12}$$

où on rappelle que $\mathbf{B}_{j,k}$ est la matrice de taille d qui ne contient que des 0 et un 1 en ligne j et colonne k. Après que les n_{qb} derniers qubits sont passés dans la porte à identifier, l'état final qui sera mesuré est noté $\boldsymbol{\rho}_\epsilon$ car (comme on va l'expliquer), cette matrice d'état caractérise entièrement ϵ :

$$\boldsymbol{\rho}_\epsilon = \frac{1}{d} \sum_{j,k=1}^{d} \mathbf{B}_{j,k} \otimes \epsilon(\mathbf{B}_{j,k}) \tag{1.13}$$

Si on estime $\boldsymbol{\rho}_\epsilon \in \mathbb{H}_{d^2}$ avec de la tomographie d'état pur, il est possible d'isoler les $\{\epsilon(\mathbf{B}_{j,k})\}_{j,k}$, en effet, par construction, le bloc $d \times d$ en haut à gauche de $\boldsymbol{\rho}_\epsilon$ est $\frac{1}{d}\epsilon(\mathbf{B}_{1,1})$, le bloc juste à droite est $\frac{1}{d}\epsilon(\mathbf{B}_{1,2})$ etc. Or, si on connaît les $\{\epsilon(\mathbf{B}_{j,k})\}_{j,k}$, alors on connaît ϵ sur une base. Donc on connaît ϵ (car le processus est linéaire). On peut montrer que, si on reprend la définition de la matrice de processus de (1.8) en décomposant les opérateur de Kraus dans la base canonique ($\widetilde{\mathbf{K}}_{j_1+(d-1)j_2} = \mathbf{B}_{j_1,j_2} \forall j_1, j_2 \in \{1, ..., d\}$), alors $\boldsymbol{\rho}_\epsilon$ est la matrice de processus $\boldsymbol{\chi}$ de (1.8) (voir [BKD14] après l'équation (8)).

L'idée est très riche et se base en fait sur l'isomorphisme de Choi–Jamiołkowski (qui associe $\boldsymbol{\rho}_\epsilon$ à ϵ). Mais en pratique, la qualité de l'estimation va dépendre de la qualité des portes CNOT utilisées (ce qui peut être gênant si on veut faire un circuit qui vérifie la qualité de portes CNOT que l'on cherche à fabriquer par exemple) et le problème de la tomographie de l'état final n'est pas trivial.

1.5.2 Caractérisation directe du processus sans QST

En 2006, Mohseni et Lidar ont observé (dans [ML06]) que les algorithmes de tomographie de processus nécessitaient toujours l'utilisation de QST, et par ailleurs, ils nécessitent $O(d^4)$ types de mesures. Ils voient le premier point (utilisation de la QST) comme un problème, car les méthodes qui utilisent la QST "results in an inherent redundancy of physical resources associated with the estimation of some superfluous parameters". Pour remédier à ces problèmes, ils proposent de faire des mesures sur l'état représenté par l'état associé à ϵ par l'isomorphisme de Choi-Jamiołkowsky : $\boldsymbol{\rho}_\epsilon$ (à l'aide du circuit de la figure 1.1) pour estimer les coefficients de la matrice densité. Tous les coefficients peuvent être identifiés avec $O(d^2)$ types de mesures à la place de $O(d^4)$. Si l'utilisateur n'est intéressé que par certains coefficients de la matrice densité, il est possible d'utiliser moins de types de mesures. Ce gain massif comparé à la méthode de Chuang et Nielsen [CN97] est dû au fait que Mohseni et Lidar utilisent des mesures beaucoup plus efficaces. La SQPT de Nielsen et Chuang ou l'AAPT avec des mesures de Pauli utilisent des mesures à 2 résultats possibles[3] alors que Mohseni et Lidar proposent des mesures à d^2 résultats possibles (d^2 est la dimension du système avec les qubits ancillaires). Chaque mesure apporte beaucoup plus d'information sur le système. D'autres travaux ont été réalisés sur cette méthode [WZH+07] [GBMK13].

1.6 Tomographie de processus parcimonieuse

1.6.1 Modèle

Depuis les années 2010, des algorithmes parcimonieux sont appliqués à la tomographie de processus quantiques [SKM+11], [FGLE12], [RVB+14], [TSK+20]. Ces méthodes sont très populaires. L'idée est de faire l'hypothèse que le processus $\boldsymbol{\epsilon}$ est parcimonieux dans le sens où la matrice de processus $\boldsymbol{\chi}$ (dont les coefficients sont définis dans (1.8)) est de rang faible. On peut montrer que, avec le système classique de la SQPT (c'est-à-dire le système qui applique le processus à identifier à des états d'entrée connus), les probabilités empiriques des mesures s'expriment de façon linéaire en fonction de la matrice de processus $\boldsymbol{\chi}$. Cette relation linéaire est dans l'équation (4) dans [RVB+14] par exemple, avec nos définitions, et si on considère que l'on réalise les mêmes nombres n_t de mesures sur tous les états de sortie, cette relation linéaire s'écrit :

$$p^{th}_{j,k}(\boldsymbol{\chi}) = \sum_{\alpha=1}^{d^2}\sum_{\beta=1}^{d^2} tr(\mathbf{e}_j\mathbf{e}_j^*\mathbf{A}_\alpha\boldsymbol{\rho}_k^{in}\mathbf{A}_\beta^*)\chi_{\alpha,\beta} \tag{1.14}$$

Où $p^{th}_{j,k}$ est la j-ième probabilité de mesure, $j \in \{1,...,dn_t\}, j = j_1 + (j_2-1)d, j_1 \in \{1,...,d\} j_2 \in \{1,...,n_t\}$ (associée au j_1-ième résultat possible du j_2-ième type de mesure) sur le k-ième état d'entrée $\boldsymbol{\rho}_k^{in}$. Le vecteur $\mathbf{e}_j$ de taille d est la transconjuguée de la j_1-ième ligne de la matrice de mesure $\mathbf{E}$ associée au j_2-ième type de mesure que l'on réalise. Les matrices $\{\mathbf{A}_\alpha\}_{\alpha=\{1...d^2\}}$ forment une base orthonormale sur $\mathbb{C}$ de $\mathbb{C}^{d\times d}$. Et $\chi_{\alpha,\beta}$ est l'élément en position (α,β) de la matrice de processus $\boldsymbol{\chi}$ associée à la base des $\mathbf{A}_\alpha$.

L'approche parcimonieuse consiste donc à trouver une matrice $\boldsymbol{\chi}$ de rang le plus faible possible qui colle le mieux possible aux mesures. Cette approche est similaire à la tomographie d'état parcimonieuse (car l'isomorphisme de Choi–Jamiołkowski conserve le rang), et dans les deux cas, le nombre de paramètres à estimer (et donc le nombre de mesures à faire) diminue fortement quand on considère des processus de rang plus faible.

[3] Il convient de noter que Chuang et Nielsen sont assez vagues sur l'algorithme de QST à utiliser et les types de mesures à réaliser car ce n'est pas le sujet de [CN97].

1.6.2 Implémentations

L'approche du premier article sur la QPT parcimonieuse [SKM+11] est un peu différente de ce que nous venons de présenter. En plus de supposer que ϵ est parcimonieux, on suppose qu'on connaît une base dans laquelle c'est le cas et que l'on peut faire des mesures dedans. Cette approche semble moins populaire récemment car les hypothèses sont plus strictes.

Les papiers suivants [FGLE12], [RVB+14], [TSK+20] se servent de l'équation (1.14) qui donne le modèle linéraire pour les mesures théoriques, les probabilités empiriques sont notées $p^{exp}_{j,k}$. un algorithme parcimonieux cherche une matrice $\boldsymbol{\chi}$ qui fait correspondre $p^{th}_{j,k}(\boldsymbol{\chi})$ à $p^{exp}_{j,k}$ (attache aux mesures) et est de faible rang (parcimonie). Il existe de nombreuses méthodes pour concilier ces deux critères contradictoires (c'est le cœur de la parcimonie/"compressed sensing"). L'approche la plus populaire pour la QPT est de résoudre le problème suivant :

$$\widehat{\boldsymbol{\chi}} = \underset{\boldsymbol{\chi}\in\mathbb{H}^+_{d^2}(\mathbb{C}) \text{ t.q. } ||p^{exp}_{j,k}-p^{th}_{j,k}(\boldsymbol{\chi})||_2\leq\varepsilon_p}{\arg\min} \operatorname{rang}(\boldsymbol{\chi}) \tag{1.15}$$

Où ε est un paramètre à régler, plus il est grand, plus le rang de la matrice obtenue sera faible. Bien entendu, minimiser le rang est très compliqué, la norme nucléaire (somme des valeurs singulières) est utilisée comme proxy, et comme $\boldsymbol{\chi}$ est une matrice hermitienne positive, la norme nucléaire est la trace :

$$\widehat{\boldsymbol{\chi}} = \underset{\boldsymbol{\chi}\in\mathbb{H}^+_{d^2}(\mathbb{C}) \text{ t.q. } ||p^{exp}_{j,k}-p^{th}_{j,k}(\boldsymbol{\chi})||_2\leq\varepsilon_p}{\arg\min} \operatorname{tr}(\boldsymbol{\chi}) \tag{1.16}$$

Il reste quelques choix à faire (au delà du choix de la valeur de ε_p), comme le choix des états d'entrée et des types de mesures à réaliser.

1.6.3 Variantes

Dans [FGLE12], Flammia et al. proposent de faire des mesures de Pauli sur la transformation par le processus d'un état pur représenté par un vecteur propre choisi aléatoirement d'une matrice de Pauli elle aussi choisie aléatoirement. Ils montrent que c'est l'équivalent de faire de la tomographie d'état parcimonieuse sur l'état $\boldsymbol{\rho}_\epsilon$ associé à $\boldsymbol{\rho}$ par l'isomorphisme de Choi-Jamiołkowski.

Dans [BKD14], les auteurs proposent de résoudre (1.16) avec des états d'entrée adaptés aux processus unitaires (nous allons décrire ces processus précisément plus loin). L'idée est que, si le processus est proche d'un processus unitaire, on peut l'estimer avec seulement d états d'entrée plutôt que les d^2 éléments d'une base des matrices hermitiennes de taille d.

Dans [KTA+20] et [TSK+20], les auteurs proposent une approche différente, le nombre d'états d'entrée et de mesures dont leur algorithme aura besoin est inconnu au début de la QPT. En traitant les premières mesures, on décide sur quels nouveaux états d'entrée on va faire des mesures (et quels types de mesures on va réaliser). L'algorithme s'arrête quand il réalise qu'il n'y a qu'un seul processus qui est cohérent avec les mesures qu'il a faites. Cet algorithme est plus général que les autres approches parcimonieuses, [KTA+20] est d'ailleurs assez critique vis-à-vis des méthodes parcimonieuses : "In practice however, this concept is only as reliable as the accuracy of the rank knowledge, and lacks an independent verification method to check the reconstruction results without fidelity comparison with target processes [RVB+14] [SKM+11]. Existing remedies for tackling these issues in compressed sensing are generally ad hoc and incomplete [SKD+17].". Nous l'avons inclus ici, car plus le processus est de rang faible plus le nombre d'états d'entrée dont on va avoir besoin diminue. Comparé aux algorithmes que nous allons introduire, il a l'inconvénient que l'opérateur ne peut pas savoir à l'avance avec quels états et quelles mesures il va travailler, et ceux-ci sont potentiellement intriqués.

1.6.4 Nos objections

Cette approche parcimonieuse n'est pertinente que si on a des raisons de penser que le processus à identifier est parcimonieux (faible rang). Les portes quantiques que l'on cherche à réaliser sont presque toujours unitaires, et on peut montrer qu'un processus unitaire est de rang 1. De notre point de vue, la raison pour laquelle un processus non unitaire proche (au sens de la norme de Frobenius des matrices de processus) d'un processus de rang 1 serait de faible rang n'est pas claire.

Pour toute matrice de rang 1, on peut trouver une matrice de rang plein arbitrairement proche (au sens de la norme de Frobenius). Et une erreur aléatoire de réalisation n'a (a priori) aucune raison de préserver le caractère parcimonieux de la matrice de processus. Pour nous, cette approche ne donne un résultat satisfaisant dans le cas d'une erreur quelconque que si la matrice de processus trouvée est bien de rang 1, il est donc plus intéressant d'imposer au processus trouvé d'être de rang 1 (et c'est ce que nous faisons). Si on trouve une matrice de rang 2 avec deux valeurs singulières de valeurs égales par exemple, on a une solution qui est bien parcimonieuse, mais pas du tout satisfaisante, car elle n'est pas unitaire, et ont peut montrer que sa distance (de Frobenius) à la matrice de rang 1 (qui correspond à un processus unitaire) la plus proche est la moité de sa norme de Frobenius (elle est donc très loin d'être unitaire). Nous pensons que la prédominance de ces méthodes dans la littérature ces dernières années est surtout due à un effet de mode des approches parcimonieuses en général dans les années 2010. Le cœur des approches parcimonieuses consiste à trouver un "point coude" (sur le graphe qui a le rang de l'estimée en abscisse et la distance entre le modèle et les données mesurées en ordonnée) qui est un bon compromis entre "rang faible" et "attache aux données". Mais dans notre cas, on sait où ce point sera (rang 1). La porte que l'on veut réaliser est de rang 1. La vraie porte est probablement de rang plein, mais toutes les valeurs singulières à part la première devraient être proches de zéro.

La proximité avec un processus unitaire n'est pas le seul argument des défenseurs des approches parcimonieuses. Dans [SKM+11], les auteurs ajoutent "The near sparsity is due to few dominant system environment interactions. This is more apparent for weakly decohering systems" avant de citer [MR09] et [KK09]. L'argument est que l'erreur d'implémentation sur les processus préserve le caractère parcimonieux du processus même si elle ne préserve pas le rang. Il existe des configurations dans lesquelles c'est vrai, et [MR09] les détaille sur le plan théorique, [KK09] contient des données expérimentales, mais ne mentionne pas le rang de la matrice du processus. Dans [FGLE12], Flammia et al. affirment que : "our method simply assumes that the noise is described by a process matrix that is low rank ; this can be rigorously justified for any noise process that involves only local interactions or few-body processes", il est vrai qu'il existe des modèles de systèmes ouverts qui donnent un rang faible à $\boldsymbol{\chi}$. Cela signifie que, si ces modèles sont exacts, il existe sans doute deux "points coudes".

1.7 Tomographie de processus inspirée de la SGQT

La SGQT (voir section 1.2.1) permet d'identifier des états quantiques, mais récemment, dans [HTF+20], l'idée de la SGQT a été adaptée à la tomographie de processus unitaire.

L'isomorphisme de Choi–Jamiołkowski est utilisé, et le système à réaliser est représenté en figure 1.2.

Ce circuit permet d'identifier le processus unitaire représenté par la matrice unitaire $\mathbf{M}$, en ajustant $\widehat{\mathbf{M}}$. On définit l'état de sortie $\boldsymbol{v}_s$ (qui est mesuré) et l'état intermédiaire $\boldsymbol{v}_a$ en sortie des portes CNOT et avant la porte à identifier (représentée par $\mathbf{M}$) défini par (1.11) que nous

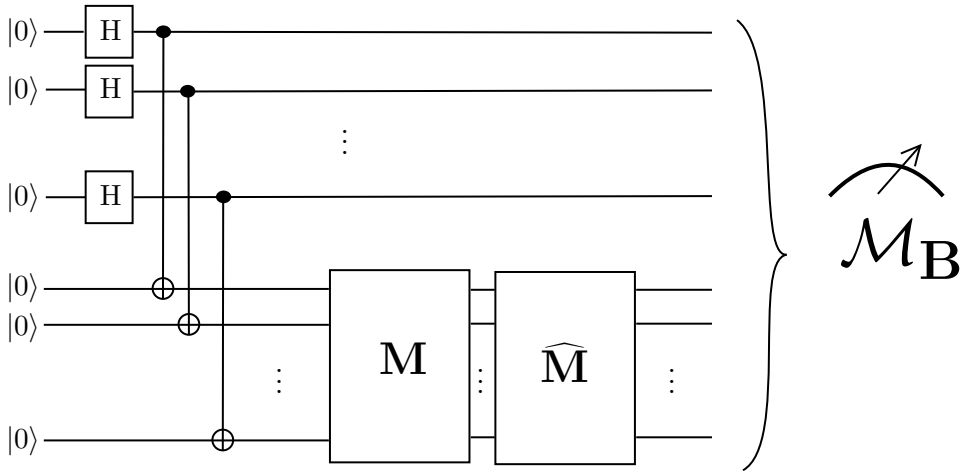


FIGURE 1.2 : Circuit de la tomographie de processus inspirée de la SGQT

réécrivons ici :

$$\boldsymbol{v}_a = \frac{1}{\sqrt{d}} \sum_{j=1}^{d} \boldsymbol{\delta}_j \otimes \boldsymbol{\delta}_j \tag{1.17}$$

On a $\boldsymbol{v}_s = \boldsymbol{v}_a$ si et seulement si $\mathbf{M}\widehat{\mathbf{M}} = \mathbf{I}_d$ car l'état de sortie caractérise uniquement la combinaison $\mathbf{M}\widehat{\mathbf{M}}$ (l'état et le processus sont liés par l'isomorphisme de Choi–Jamiołkowski) donc si $\mathbf{M}\widehat{\mathbf{M}}$ ne change pas l'état d'entrée, alors, la combinaison $\mathbf{M}\widehat{\mathbf{M}}$ est l'identité, et donc $\mathbf{M} = \widehat{\mathbf{M}}^*$.

Or, si on s'autorise des mesures intriquées, alors, tester une hypothèse d'égalité sur un état est une des opérations les plus faciles à réaliser avec une mesure quantique. Pour cela, on choisit comme type de mesure $\mathcal{M}_{\mathbf{B}}$ la projection sur une base orthogonale $\mathbf{B}$ dont le premier vecteur (associé à 0...0) est $\boldsymbol{v}_a$. Avec cette mesure, si (et seulement si) $\boldsymbol{v}_s = \boldsymbol{v}_a$ alors on mesure le premier résultat possible $(0, ..0)$ avec une probabilité 1. En pratique le type de mesures de [HTF+20] n'a que 2 résultats possibles et pas d. Le premier résultat correspond à notre 0...0, et le deuxième correspond à tous les autres (c.à.d. la probabilité d'avoir le deuxième résultat est la somme des probabilités des $d-1$ résultats de notre type de mesure). Ces deux mesures ne sont pas équivalentes, mais, avec l'usage qu'on en fait (on ne compte que le premier résultat possible) elles sont tout autant adéquates.

Avec le même algorithme d'optimisation statistique que pour la SGQT, on peut modifier les paramètres de $\widehat{\mathbf{M}}$ jusqu'à ce que les mesures que l'on fait sur l'état de sortie nous indiquent que $\mathbf{M} = \widehat{\mathbf{M}}^*$. Cette idée pose de nombreux problèmes avec les cas d'utilisation que nous envisageons, nous les donnons ici par ordre d'importance :

- Le dispositif suppose qu'il est possible de créer n'importe quelle porte unitaire connue $\widehat{\mathbf{M}}$ et l'estimation de $\mathbf{M}$ sera mauvaise si on maîtrise mal $\widehat{\mathbf{M}}$. Il ne peut donc être appliqué que si on est déjà capable de générer toute les portes unitaires sur n_{qb}, nous ne faisons pas cette hypothèse.
- La mesure $\mathcal{M}_{\mathbf{B}}$ est intriquée.
- Le dispositif suppose qu'il est possible de réaliser les portes CNOT et les portes de Hadamard, et la qualité de l'estimation de $\mathbf{M}$ dépend de la qualité de ces portes.

La SGQT adaptée à la tomographie de processus pourrait être défendue en mettant en avant le fait que dans [HTF+20], elle a été validée expérimentalement. Cependant, l'expérience de

[HTF+20] a été réalisée pour des processus unitaires sur un seul qubit. Ces processus sont particulièrement simples, ils n'ont que 3 paramètres réels (que l'on doit faire varier pour changer $\widehat{\mathbf{M}}$), et ils ne créent pas d'intrication entre plusieurs qubits, car il n'y a qu'un seul qubit. Le passage à deux qubits nécessite de pouvoir créer précisément n'importe quelle porte quantique d'intrication à 15 paramètres, cela nous semble très compliqué.

1.8 Tomographie d'un ensemble de portes (GST)

La tomographie d'un ensemble de portes ("gate set tomography" en anglais ou GST) [NGR+21a] [MGS+13] [BKGN+13] [NGR+21b] part du constat que :

- Pour faire de la tomographie de processus, il faut toujours connaître les états d'entrée et/ou avoir bien caractérisé les mesures quantiques que l'on effectue sur les états.
- Pour bien connaître un état, il faut avoir caractérisé la porte qui a permis de le préparer ou alors faire de la tomographie d'état avec des mesures quantique bien caractérisées.
- Pour caractériser une mesure quantique, il faut faire de la tomographie de mesure (i.e. avoir estimé la matrice unitaire $\mathbf{E}$ associée). La tomographie ne peut complètement caractériser une mesure que si on applique cette mesure à des états connus.

Les trois problèmes de tomographie (d'état, de processus et de mesure) ne peuvent donc pas être considérés indépendamment, et si on veut estimer un processus, on ne peut pas considérer que les états d'entrée ou les mesures sont connus. L'idée de la GST est de considérer un unique état initial inconnu $\boldsymbol{\rho}$ auquel on peut facilement réinitialiser le système, un unique type de mesure $\mathcal{M}_{GST}$ (à deux résultats possibles) dont on ne connaît pas les paramètres et un ensemble de portes quantiques $\{G_k\}_k$ (pas forcément unitaires) à identifier. Le fait de ne considérer qu'un seul type de mesure est justifié parce que, pour certaines architectures matérielles, on ne mesure les états que dans la base de référence, et on peut simuler des types de mesures différents en appliquant une porte unitaire sur les états que l'on veut mesurer. Mesurer un état $\boldsymbol{v}$ dans la base de référence après l'avoir fait passer dans une porte quantique représentée par la matrice unitaire $\mathbf{E}$ revient à mesurer l'état dans la base $\mathbf{E}$. Dans la suite de ce document, on appelle cela une mesure par porte interposée par opposition à une mesure directe. On réalise plusieurs expériences, les portes quantiques à identifier sont appliquées dans des ordres différents, mais toujours après l'initialisation de l'état et avant la mesure.

Avec la GST, on peut estimer certains paramètres des portes sans aucune connaissance a priori sur les mesures ou l'état d'entrée. Cependant, sans aucune référence, il est impossible d'avoir tous les paramètres. Même dans des conditions optimales, le résultat obtenu ne sera bon qu'à d'importantes indéterminations près (appelées gauge). Nous considérons que ces indéterminations sont rédhibitoires pour les cas d'application que nous envisageons. Par exemple, imaginons que l'on veut utiliser la GST pour estimer une porte (unitaire) CNOT représentée par la matrice $\mathbf{U}_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$. Quelle que soit la matrice unitaire perturbatrice $\mathbf{R}$ et les portes unitaires $G_{\mathbf{R}}$ et $G_{\mathbf{R}^*}$ associées à $\mathbf{R}$ et sa transconjuguée $\mathbf{R}^*$, les mesures que l'on fait dans le cadre de la GST en appliquant les $\{G_k\}_k$ (une d'entre elles est la porte CNOT) à ρ avant de les mesurer avec $\mathcal{M}_{GST}$ sont indiscernables des mesures que l'on ferait avec l'état de base $\mathbf{R}^*\rho\mathbf{R}$, les portes $\{G_{\mathbf{R}} \circ G_k \circ G_{\mathbf{R}^*}\}_k$ (en appliquant $G_{\mathbf{R}}$ avant chaque porte et $G_{\mathbf{R}^*}$ après, $\circ$ est la composition des fonctions) et la mesure $G_{\mathbf{R}^*} \circ \mathcal{M}_{GST}$ (qui applique $G_{\mathbf{R}^*}$ avant de mesurer les états avec $\mathcal{M}_{GST}$). Donc si on veut se servir de la GST pour estimer la matrice unitaire représentant

une porte CNOT, alors, même dans des conditions idéales, on peut obtenir n'importe quelle matrice de $\{\mathbf{R}^*\mathbf{U}_{CNOT}\mathbf{R}\}_{\mathbf{R}\in\mathbb{U}_d(\mathbb{C})}$. En pratique, on va trouver une matrice qui a les bonnes valeurs propres (qui sont des phases pour des matrices unitaires), mais les vecteurs propres (qui forment une base orthogonale pour une matrice unitaire) sont multipliés par une matrice unitaire (cette multiplication fait que l'on n'a aucune information sur les vecteurs propres). Par exemple, avec $R = \begin{pmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & \frac{-1}{2} \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{2} & \frac{1}{2} \end{pmatrix}$, on peut avoir $\mathbf{R}^*\mathbf{U}_{CNOT}\mathbf{R} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ comme estimée de $\mathbf{U}_{CNOT}$, on ne peut donc pas différencier la porte CNOT d'une porte qui fait juste des déphasages. Ce n'est pas du tout satisfaisant, et contraste avec le jusqu'au boutisme de certains partisans de la GST, d'après Blume-Kohout et al. dans la conclusion de [BKGN+13], par exemple "We do not expect that gate set tomography will be another kind of tomography, standing shoulder to shoulder with state tomography, process tomography, and measurement tomography. It is intended to *replace* them". Nous pensons qu'il y a toujours une place pour la tomographie de processus pour identifier une seule porte qui va ensuite être utilisée dans des circuits différents. Comme le nom l'indique cependant, la GST prend plus de sens si on veut estimer plusieurs portes en même temps, chaque porte sera mal connue indépendamment des autres, mais les erreurs de gauge sur chaque porte sont très similaires (la matrice $\mathbf{R}$ est la même sur toutes les portes à estimer). L'introduction de la GST a aussi pour mérite de mettre davantage en évidence les problèmes de la plupart des algorithmes de tomographie de processus qui ont une confiance aveugle en les états d'entrée et les mesures.

1.9 Certification d'une porte sans estimation de l'erreur

Ces approches partent du constat que, très souvent, les réalisations de la QPT demandent beaucoup de mesures et de calculs pour estimer les paramètres de Kraus, et finalement, ils ne donnent que la fidélité (une métrique de proximité avec la porte que l'on souhaite réaliser), on estime donc $O(d^4)$ paramètres pour finalement calculer un seul scalaire.

Plutôt que d'aider à la fabrication d'une porte quantique en identifiant précisément les erreurs afin d'ajuster la façon dont la porte est fabriquée, la "Monte Carlo process certification" et le "randomized benchmarking" permettent de vérifier qu'une porte (ou un ensemble de portes) que l'on va utiliser sur un circuit quantique ne contient aucune porte défaillante, et si ce n'est pas le cas, ils donnent une idée de l'erreur à laquelle on peut s'attendre.

1.9.1 Monte Carlo process certification

La "Monte Carlo process certification" vise à estimer la fidélité d'une porte qui est censée être unitaire. L'idée est d'utiliser des mesures de Pauli choisies de façon aléatoire avec une loi qui dépend du processus cible et qui donne de plus grosses probabilités aux mesures qui apportent plus d'information sur la fidélité. Ces mesures sont ensuite utilisées pour estimer la fidélité avec un estimateur qui converge en $O\left(\frac{1}{\sqrt{n_t}}\right)$ où n_t est le nombre de types de mesures. Le seul problème que nous voyons avec cette idée est qu'elle utilise des mesures de Pauli multi-qubit à 2 résultats possibles. Nous expliquons dans la section 2.1.2 pourquoi ce n'est pas optimal.

1.9.2 Randomized benchmarking

Le "randomized benchmarking" [EAŻ05], [KLR+08], [MGE11], [MGE12], [OWE19] est une autre méthode d'estimation de la fidélité. L'implémentation exacte varie selon les versions, mais l'idée

de base est de considérer plusieurs portes (en général quelques dizaines) que l'on va utiliser dans un circuit quantique, et de les appliquer dans un ordre aléatoire à un état d'entrée connu (généralement tous les qubits à $|0\rangle$). Le circuit aléatoire est modifié pour que, si toutes les portes sont parfaites, l'état final soit un état de la base de référence (par exemple tous les qubits à $|0\rangle$). Pour vérifier que les portes fonctionnent comme attendu, on va donc mesurer l'état final dans la base de référence et vérifier que l'on obtient bien la mesure associée à l'état final que l'on devrait avoir.

Cette méthode est particulièrement adaptée pour estimer les portes de Clifford (définies en section 1.1.6) car ces portes sont souvent idempotentes (Hadamard, CNOT) ou si ce n'est pas le cas, elles sont équivalentes à l'identité si on les applique un nombre connu de fois. Des ajustements peuvent être faits pour que la méthode soit résistante aux erreurs systématiques sur les états initiaux et les mesures (voir [MGE12]).

Dans [MGE12], Kimmel et al. proposent d'utiliser l'idée du "randomized benchmarking" pour obtenir des informations sur une porte. L'idée est de calculer la fidélité de la porte à identifier par rapport à plusieurs portes de Clifford et d'en déduire des informations sur la porte.

1.10 Autres travaux

Dans [OSB15a] et [OSB15b] Omkar et al. proposent d'utiliser des codes correcteurs d'erreurs quantiques pour estimer un processus. Les codes correcteurs quantiques fonctionnent comme les codes correcteurs numériques en télécommunication, les états sont initialisés dans un ensemble code, on met des états dans l'ensemble $\mathcal{E}$ de l'espace de Hilbert. Quand on reçoit les qubits, on sait qu'ils doivent être dans l'ensemble $\mathcal{E}$ et s'ils n'y sont pas, on les projette sur cet ensemble. L'idée de [OSB15a] et [OSB15b], est de considérer les états initiaux dans l'ensemble $\mathcal{E}$, et de laisser le processus que l'on veut identifier les modifier. En mesurant les états de sortie, et en calculant leur distance à $\mathcal{E}$, on peut estimer l'impact du processus avec des hypothèses limitées sur les états d'entrée (on suppose juste qu'ils sont dans $\mathcal{E}$).

Dans, [TEŘH11], les auteurs proposent un algorithme pour estimer des processus que l'on a évalués sur trop peu d'états d'entrée. Pour ce faire, ils proposent de maximiser simultanément la vraisemblance des mesures et l'entropie de l'état associé au processus par l'isomorphisme de Choi-Jamiołkowski.

Dans [PKB+20], les auteurs proposent d'utiliser des réseaux de neurones pour la QPT. Ils partent du même principe que les défenseurs de la GST, i.e. on ne peut se fier ni aux états préparés ni aux mesures. Ils modélisent des mesures qui prennent en compte différents types d'erreurs et entraînent des réseaux de neurones pour identifier le processus.

Dans [XLW+22], les auteurs estiment un processus unitaire avec un circuit quantique paramétrique (CQP voir [BLSF19]). Les paramètres du CQP sont ajustés pour que ce dernier coïncide avec le processus que l'on veut identifier sur les états tests générés aléatoirement. L'idée est assez similaire à celle de la SGQT.

1.11 Tomographie aveugle de processus

En 2015, Yannick et Alain Deville ont introduit la version aveugle de la QPT (BQPT) dans [DD15], puis ils l'ont détaillée dans [DD17c], [DD17b], [DD17a] et plus récemment dans [DD20]. Dans ces articles, les auteurs se sont concentrés sur la tomographie du processus de couplage d'Heisenberg à deux qubits à symétrie cylindrique. [DD17c], [DD15] et [DD17b] sont inspirés de la séparation de sources indépendantes [CJ10]. Les qubits non-intriqués sont pensés comme des sources indépendantes. Ces algorithmes sont dits aveugles, car les états d'entrée sont considérés comme inconnus, cela supprime le problème des erreurs de préparation.

Dans [DD17a], le processus de couplage d'Heisenberg est identifié avec une configuration comparable avec celle de la SGQPT (sans qubits ancillaires) avec des états d'entrée qui sont modifiés par le processus en sortie duquel on applique un processus quantique "réglable", et on détecte pour quelle valeur du processus "réglable" les états de sortie sont non-intriqués. On en déduit que le processus à identifier vaut l'inverse du processus "réglable" en fin d'expérience. Le fait d'avoir besoin du processus "réglable" n'est pas aussi problématique que pour la SGQPT car le processus étudié (couplage d'Heisenberg à deux qubits) est particulièrement simple (4 paramètres à régler). Dans [DD17c], d'autres options sont explorées, elles permettent de se passer de ce processus quantique "réglable".

L'itération la plus récente de la BQPT [DD20] repose sur une idée différente. Les auteurs utilisent des propriétés statistiques des états d'entrée pour estimer les paramètres du processus à partir de moments des mesures sur les états de sortie. Ces moments peuvent être estimés sans que l'on ait besoin de mesurer plus d'une copie de chaque état d'entrée (on parle de "one-shot measurements") ce qui simplifie considérablement la préparation. Pour lever des ambiguïtés, il faut mesurer l'action du processus appliqué deux fois aux états d'entrée, cette idée a été reprise dans la présente thèse.

La présente thèse avait commencé avec, pour objectif de construire sur ces idées de BQPT et de les étendre des processus de couplage d'Heisenberg aux processus unitaires. Nous avons cependant choisi d'étudier des algorithmes assez différents qui font les mêmes hypothèses sur les états d'entrée, mais se basent entièrement sur le fait que le processus à identifier est appliqué au moins deux fois pour l'identifier.

1.12 Tomographie de processus unitaire

Pour cette thèse, nous avons déterminé qu'il était avantageux de chercher le processus qui correspond le mieux aux mesures dans l'ensemble des processus unitaires plutôt que dans l'ensemble des processus (CPTPM). Dans la section 1.12.1, nous justifions ce choix, et dans les sections suivantes, nous présentons les travaux d'autres auteurs qui ont fait le même choix.

1.12.1 Motivation

Presque toutes les portes/opérations quantiques usuelles sont unitaires. La principale exception à cette règle est l'assignation (assigner un qubit à $|0\rangle$ par exemple), et plutôt que d'estimer les $d^4 - d^2$ paramètres de ces assignations, il est plus pertinent de vérifier que les états de sorties sont assignés correctement.

Nous pensons qu'il est pertinent d'identifier les processus unitaires avec des algorithmes adaptés (qui ne cherchent que dans l'ensemble des matrices unitaires) pour les raisons suivantes :

- L'estimation des d^2-1 paramètres de la matrice unitaire associée (à une phase globale près) au processus unitaire (nous le montrerons en section 3.3), nécessite considérablement moins de mesures que d'estimer les $d^4 - d^2$ paramètres de Kraus. Les algorithmes de QPT non-unitaires ont généralement besoin d'estimer les paramètres de d^2 états mélanges différents (en sortie du processus) pour estimer tous les paramètres d'un processus non-unitaire. La QPT unitaire de Baldwin et al. [BKD14] n'estime que les paramètres de d états purs différents.

- Il est très simple de comparer deux matrices unitaires (représentant le processus que l'on voudrait réaliser et le processus unitaire estimé) qui devraient être égales à une phase près, il suffit d'en rephaser un avec la bonne phase globale. Et l'information apportée est riche : si les portes sont réalisées dans un système fermé avec un hamiltonien constant,

on sait exactement comment l'hamiltonien (ou Δ_t) doit être modifié pour que la porte soit conforme au processus que l'on veut réaliser. Alors que de savoir que (et comment) le système interagit avec l'environnement nous indiquera simplement qu'il vaut mieux isoler le système, ce que l'on essaye toujours de faire autant que possible de toute façon.

- La représentation avec les paramètres de Kraus, ou avec la matrice de processus est très difficile à interpréter. La principale information qui peut en être extraite est la fidélité de la porte, ce n'est pas différent de ce qu'apporte la "Monte Carlo process certification" ou le "randomized benchmarking".

- Faire l'hypothèse que l'on sait réaliser des portes unitaires nous permet de considérer que tous les états mesurés sont purs. En effet, il est raisonnable de supposer que la première étape de génération d'états : l'assignation de tous les qubits à $|0\rangle$ est au moins répétable (donc génère un état pur), et si toutes les portes utilisées pour modifier cet état sont unitaires, alors l'état restera pur (les portes unitaires préservent la pureté des états). L'avantage des états purs est qu'ils ont moins de paramètres ($2d-2$ paramètres contre d^2-1), ils peuvent donc être estimés de façon plus exacte avec moins de types de mesures

Il est vrai que les erreurs que l'on cherche à identifier avec la QPT peuvent rendre le processus non-unitaire, et que si cette erreur est assez grande, notre estimée unitaire est perturbée. Mais chercher à estimer les interactions avec l'environnement en multipliant le nombre de paramètres par d^2 alors que ces interactions sont susceptibles de changer avec l'environnement nous semble être une mauvaise idée qui fait drastiquement augmenter le nombre de mesures à réaliser pour une précision moindre et un gain douteux. Si on suspecte que le système n'est pas bien fermé et que les interactions avec l'environnement ne peuvent pas être négligées, nous pensons qu'une approche similaire au "randomized benchmarking" (estimer l'écart à la cible plutôt que tous les paramètres) est plus adapté que la QPT pour $n_{qb} > 1$. De plus, nos algorithmes peuvent aussi estimer la vraisemblance des mesures réalisées dans le modèle unitaire, ce qui peut donner une indication quant à la non-unitarité du système.

De notre point de vue, les raisons pour lesquelles les systèmes non-unitaires sont beaucoup plus étudiés que les processus unitaires dans la littérature malgré les arguments que nous avons donnés sont les suivantes :

- Dans la littérature, un algorithme plus général est souvent perçu comme supérieur (à raison ou à tort).

- La plupart des auteurs préfèrent raisonner avec des états quantiques représentés par des matrices densité, car (i) ils sont plus généraux que les états purs, (ii) les mesures dépendent linéairement des coefficients des matrices densité des états mélanges (1.2). Et les états mélanges, contrairement aux états purs, forment un ensemble stable par l'application d'un processus non-unitaire.

- Avec un processus représenté par une matrice de processus $\boldsymbol{\chi}$ et des états d'entrée représentés par des matrices densité, on a une relation linéaire (1.14) entre les probabilités des mesures et les éléments de la matrice $\boldsymbol{\chi}$. Après avoir inversé (1.14), il ne reste alors plus qu'à s'assurer que les contraintes de positivité et de préservation de la trace sont satisfaites. Imposer au processus d'être unitaire ajoute une contrainte (de rang) compliquée à (1.14), plutôt que de garder cette contrainte de rang, les adeptes de la tomographie de processus parcimonieuse ont préféré utiliser la norme nucléaire comme sortie. Nous avons expliqué nos réserves vis-à-vis de cette idée dans la section 1.6.4.

1.12.2 Conditions d'identifiabilité

Dans leur article de 2013, [RGK13] Reich et al. étudient la tomographie de processus unitaire sans vraiment proposer un algorithme de QPT. Ils étudient plutot quelles conditions doivent être imposées aux états (mélanges) d'entrée pour que le processus unitaire soit identifiable parmi l'ensemble des processus quantiques. Dans ce contexte, l'identifiabilité signifie qu'il n'existe aucun autre processus quantique (CPTPM) qui produise les mêmes états de sortie à partir des états d'entrée.

La condition nécessaire et suffisante sur les états d'entrée $\{\boldsymbol{\rho}_j^{in}\}_j$ est la suivante :

$$com(\{\boldsymbol{\rho}_j^{in}\}_j) = \{e^{i\theta}\mathbf{I}_d\}_{\theta\in\mathbb{R}}, \text{ avec } com(\{\boldsymbol{\rho}\}_j^{in}) = \{\mathbf{P} \in \mathbb{U}_d(\mathbb{C}), \text{ t.q. } \forall j, \ \mathbf{P}\boldsymbol{\rho}_j^{in} = \boldsymbol{\rho}_j^{in}\mathbf{P}\} \tag{1.18}$$

En clair, toute matrice unitaire qui commute avec toutes les matrices de mélange des états d'entrée est l'identité à une phase près.

Les auteurs étudient comment caractériser la fidélité de la porte sur laquelle la QPT est réalisée avec des états qui satisfont (3.26). Ils proposent aussi des états (mélanges et purs) qui vérifient (3.26). En s'autorisant des états mélanges, ils ont besoin de deux états, alors qu'avec des états purs, il leur en faut $d+1$ (nous pourrions satisfaire leur condition avec d états purs). Ce n'est pas si surprenant, avec un processus unitaire, si on a en entrée un état mélange dont la matrice densité a d valeurs propres distinctes, alors la sortie a les mêmes valeurs propres et les vecteurs propres sont multipliés par la matrice unitaire du processus. En calculant la décomposition en valeurs singulières de la sortie, on peut donc savoir comment le processus agit sur les vecteurs propres (à une phase près), alors qu'avec des états purs, on ne sait que comment il agit sur un vecteur (à une phase près).

Dans [GJ14], les auteurs considèrent le circuit de la figure 1.1 (qui crée l'état associé au processus par l'isomorphisme de Choi–Jamiołkowski) et établissent un lien entre le rang de la matrice de processus et un nombre d'observables (composantes de POVMs) suffisant pour l'identifier. En particulier, ils montent que si le processus est de rang 1 (i.e. unitaire) il peut être identifié parmi tous les autres processus unitaires avec $4d^2 - 2d - 4$ observables, et parmi tous les processus quantiques avec $5d^2 - 3d - 4$ observables. [GJ14] et [RGK13] ont tous les deux été réalisés indépendamment la même année (2013). Mais, pour nous [RGK13] est plus utile, car il n'utilise pas le circuit de la figure 1.1 mais se place dans les conventions de la SQPT. De plus, si on s'autorise des POVM (comme [GJ14]) pour faire la QST, alors, les états purs de [RGK13] permettent d'identifier le processus avec les $2d-1$ observables de [FSC05] pour chacun des $d+1$ états, donc $2d^2 + d - 1$ observables en tout.

1.12.3 Travaux de Baldwin, Kalev et Deutsch

Dans leur article de [BKD14] Baldwin et al. proposent un algorithme de tomographie de processus unitaire. Leur algorithme est adapté aux d états suivants :

$$\left\{\boldsymbol{\delta}_1, \{\frac{1}{\sqrt{2}}(\boldsymbol{\delta}_1 + \boldsymbol{\delta}_k)\}_{k\in\{2,\dots,d\}}\right\} \tag{1.19}$$

(où $\boldsymbol{\delta}_k$ est le vecteur de taille d qui contient $d-1$ zéros et un 1 en position k). Ces états satisfont (3.26) et rendent la tomographie de processus unitaire très simple.

Avec la sortie associée au premier état d'entrée, on connaît (après QST) $\mathbf{M}\boldsymbol{\delta}_1$ à une phase θ_1 près ($\mathbf{M}$ est la matrice unitaire qui représente le processus). On appelle $\boldsymbol{w}_1$, l'estimée de $\mathbf{M}\boldsymbol{\delta}_1$ (on néglige l'erreur de QST) $\boldsymbol{w}_1 = \mathbf{M}\boldsymbol{\delta}_1 e^{i\theta_1}$, donc on connaît la première colonne de $\mathbf{M}$ à une phase près. Avec la sortie associée à la k-ième ($k \in \{2,\dots,d\}$) entrée, on a $\boldsymbol{w}_k = \frac{1}{\sqrt{2}}e^{i\boldsymbol{\theta}_k}(\mathbf{M}\boldsymbol{\delta}_1 + \mathbf{M}\boldsymbol{\delta}_k)$. Comme $\boldsymbol{\delta}_1 \perp \boldsymbol{\delta}_k$ et comme $\mathbf{M}$ préserve l'orthogonalité, on a $\mathbf{M}\boldsymbol{\delta}_1 \perp \mathbf{M}\boldsymbol{\delta}_k$, et donc, avec $\boldsymbol{w}_1$ et $\boldsymbol{w}_k$ (que l'on connaît), on peut calculer $\boldsymbol{w}_1^*\boldsymbol{w}_k = \frac{1}{\sqrt{2}}e^{i(\boldsymbol{\theta}_k - \boldsymbol{\theta}_0)}((\mathbf{M}\boldsymbol{\delta}_1)^*(\mathbf{M}\boldsymbol{\delta}_1) + 0) = \frac{1}{\sqrt{2}}e^{i(\boldsymbol{\theta}_k - \boldsymbol{\theta}_1)}$.

Or $\boldsymbol{w}_k\sqrt{2}e^{i(\boldsymbol{\theta}_1-\boldsymbol{\theta}_k)} = e^{i\boldsymbol{\theta}_1}(\mathbf{M}\boldsymbol{\delta}_1 + \mathbf{M}\boldsymbol{\delta}_k)$, donc $e^{i\boldsymbol{\theta}_1}\mathbf{M}\boldsymbol{\delta}_k = \frac{\boldsymbol{w}_k}{\boldsymbol{w}_1^*\boldsymbol{w}_k} - \boldsymbol{w}_1$. Donc on connaît tous les $\mathbf{M}\boldsymbol{\delta}_k$ (qui est la k-ième colonne de $\mathbf{M}$) à un facteur de phase $e^{i\boldsymbol{\theta}_1}$ près qui est le même sur toutes les colonnes, on a donc $\mathbf{M}$ à une phase près, ce qui est l'objectif de la QPT.

On n'aurait pas pu choisir les $\{\boldsymbol{\delta}_k\}_k$ comme états d'entrée car la QST des états de sortie donnerait toutes les colonnes de $\mathbf{M}$ à des facteurs de phase différents près. Et, pour $n_{qb} = 1$, on ne pourrait pas faire la différence entre $\mathbf{M} = \mathbf{I}$ et $\mathbf{M} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$. D'ailleurs, ces états ne vérifient pas (3.26) car n'importe quelle matrice unitaire diagonale commute avec les matrices de mélange associées.

Les auteurs proposent aussi un algorithme pour identifier les processus presque unitaires (near-unitary maps). Cet algorithme est inspiré du compressed sensing, nous l'avons mentionné dans 1.6. Contrairement à l'algorithme de QPT unitaire, il n'y a pas de contrainte ferme sur les états d'entrée avec cette approche, mais les auteurs recommandent d'utiliser les mêmes états que pour la tomographie unitaire.

Les algorithmes de QPT (comme les nôtres et beaucoup d'autres) introduits par les auteurs se basent sur des estimées (avec de la QST) des états de sortie. Les auteurs choisissent de considérer l'algorithme de Flammia et al. [FSC05], mais n'importe quel algorithme de QST adapté aux états purs aurait pu être considéré.

1.12.4 Identification d'hamiltonien

Une autre façon de voir la tomographie de processus est d'identifier l'hamiltonien (considéré comme constant dans le temps) $\mathbf{H}$ plutôt que la matrice unitaire $\mathbf{M} = e^{-\frac{i}{\hbar}\mathbf{H}\Delta_t}$ qui caractérise l'évolution (après avoir attendu Δ_t) des états dans le système à hamiltonien constant. Fondamentalement, le problème est le même tant que $\Delta_t < \frac{\pi}{h_d - h_1}$ où h_d et h_1 sont respectivement les plus grande et plus petites valeurs propres (réelles) de $\mathbf{H}$ [WDQ+17]. Mais les communautés d'auteurs sont très différentes, dans le domaine de l'identification d'hamiltonien, l'article le plus reconnu est sans doute [ZS14]. Les auteurs résolvent le problème sans faire de QST sur les états mesurés (c'est assez courant dans le domaine). Les auteurs considèrent aussi des a priori sur l'hamiltonien et des mesures à intervalle de temps réguliers qui ne sont pas effectuées sur tous les qubits. Les paramètres de l'hamiltonien sont estimés directement avec les observations en utilisant l'algorithme ERA ("eigensystem realization algorithm") [JP85] qui sert à identifier des système dynamiques. Les auteurs ne sont pas très clairs sur la quantité d'échantillons dont on a besoin et sur les conditions d'identifiabilité de l'hamiltonien en général.

[WDQ+17] se place dans le cadre de la QPT. Contrairement à [ZS14], aucun a priori n'est fait sur l'hamiltonien, on mesure tous les qubits, et les états mesurés sont estimés avec de la QST. L'idée est de prendre une base des matrices hermitiennes comme états d'entrée (donc d^2 états d'entrée). Comme dans [CN97], les auteurs utilisent les estimées des états pour estimer la matrice $\mathbf{\Lambda}$ de (1.14) dont ils se servent pour obtenir la matrice de processus. Puis ils utilisent les contraintes sur la matrice de processus (positive et de rang 1) pour obtenir une estimée de l'hamiltonien. Les auteurs détaillent avec soin la complexité des algorithmes qui doivent être réalisés sur ordinateur classique pour estimer l'hamiltonien à partir des mesures, ce qui n'est important qu'en grande dimension. Cependant, un défaut majeur de l'algorithme (surtout en haute dimension) est qu'il nécessite d^2 états d'entrée comparé aux $O(d)$ qui sont généralement nécessaire pour les processus unitaires.

1.12.5 Autres travaux

Dans [KLY15], les auteurs proposent un algorithme pour estimer les paramètres de deux types de portes unitaires mono-qubit. Ces deux types de portes sont dit générateurs : toute porte "cible" peut être générée en les combinant. L'identification des paramètre des portes se fait en estimant les sorties des portes quand les états $|0\rangle$ et $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ sont en entrée. Cela peut être vu comme un cas particulier du dispositif de Baldwin et al. pour un seul qubit. Comparé à l'algorithme de Baldwin et al. [BKD14], celui de [KLY15] a l'avantage d'être résistant à certains types d'erreurs, notamment les erreurs systématiques. Il a l'inconvénient de n'être défini que pour les portes mono-qubit qui, en pratique, sont généralement plus facile à réaliser que les portes multi-qubit (voir [KBGK18], "single-qubit gates have been demonstrated with high fidelity, but entangling gates have lower fidelities"). Nous ne considérons pas l'algorithme de [KLY15] comme un algorithme de QPT, il ne vise pas à estimer les paramètres d'une porte quelconque, mais propose un algorithme pour bien caractériser deux types de portes "générateurs".

Très récemment (22 juin 2023), dans [LGDM23], Lopez et al. ont proposé une configuration de QPT assez proche de la nôtre. L'idée et d'utiliser une seule valeur de l'état initial, de le placer dans un système à hamitonien constant, et de mesurer des copies de l'état initial à différent instants. Les similarités s'arrêtent là, leur algorithme est très différent du nôtre, il ne réalise pas la QST des états mesurés et estiment le processus en minimisant un critère non-convexe initialisé aléatoirement plusieurs fois dans l'espoir de trouver le minimum global. Des arguments théoriques sont utilisés pour montrer que le nombre de types de mesures réalisées et d'instants où l'état est mesuré est minimal, mais comme l'initialisation de la minimisation est aléatoire, la robustesse de l'algorithme final laisse à désirer.

Chapitre 2

Tomographie d'état

Sommaire

Le présent chapitre présente les choix que nous avons fait pour les types de mesures à réaliser et les algorithmes de tomographie d'état pur. Il s'agit d'une version plus détaillée de notre article [VD23a] paru dans Physical Review A en janvier 2023.

2.1 Mesures

2.1.1 Nos types de mesures

Nous avons décrit en section 1.1 certaines caractéristiques des types de mesures que nous allons utiliser. Ce sont des mesures projectives (pas de POVM) à d résultats possibles (soit le nombre maximal de résultats possibles) et non intriquées (la mesure se décompose en n_{qb} mesures sur chacun des qubits). La figure 2.1 présente deux visions équivalentes de la réalisation de ces mesures :

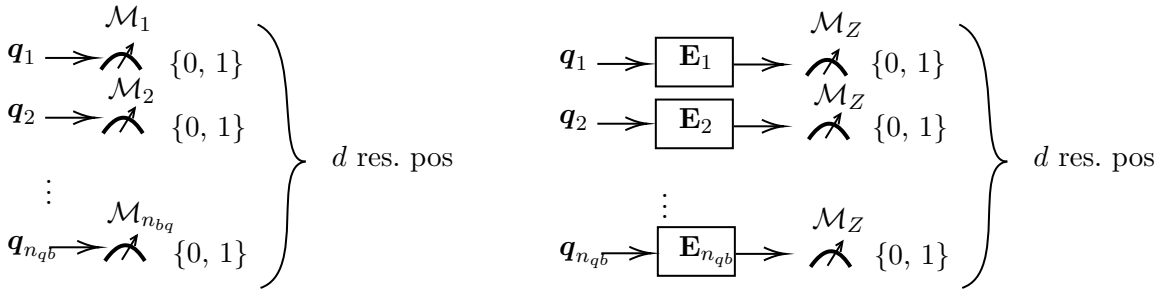


FIGURE 2.1 : Mesure projective à d résultats possibles non-intriquée. Elle peut être réalisée en mesurant directement tous les qubits dans les bases voulues (gauche) ou en les mesurant tous dans la base de référence après leur avoir appliqué des portes unitaires (droite). Chaque mesure mono-qubit a deux résultats possibles (que l'on renomme 0 et 1), en tout, il y a $2^{n_{qb}} = d$ résultats possibles.

Chaque qubit $(\boldsymbol{q}_1, ..., \boldsymbol{q}_{n_{qb}})$ est mesuré une fois avec une mesure à 2 résultats possibles, ces résultats sont renommés 0 et 1, les résultats sont ensuite concaténés pour obtenir une chaîne de n_{qb} bits (d résultats possibles) qui correspond à la mesure que l'on fait sur le système. Chaque mesure mono-qubit $\mathcal{M}_k$ ($k \in \{1, ..., n_{qb}\}$) est caractérisée par une matrice unitaire 2×2 : $\mathbf{E}_k$. Toute matrice unitaire s'écrit de la façon suivante :
$\mathbf{E}_k = \begin{pmatrix} \cos(\theta)e^{i\phi_1} & -\sin(\theta)e^{i\phi_2} \\ \sin(\theta)e^{i\phi_3} & \cos(\theta)e^{i(\phi_2+\phi_3-\phi_1)} \end{pmatrix}$ voir annexe A.1, les 4 paramètres $\theta, \phi_1, \phi_2, \phi_3$ sont des angles entre 0 et 2π. Ces 4 paramètres sont redondants dans notre cas cependant. En effet, les seules observations que l'on a sont les estimées des probabilités des deux résultats possibles, ces probabilités sont contenues dans le vecteur $|\mathbf{E}_k \boldsymbol{v}|^2$, et pour un qubit représenté par $\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 e^{i\theta_{\boldsymbol{v}}} \end{pmatrix}$ $(v_1, v_2 \in \mathbb{R}_+, \theta_{\boldsymbol{v}} \in [0, 2\pi])$:
$$\left|\mathbf{E}_k \begin{pmatrix} v_1 \\ v_2 e^{i\theta_{\boldsymbol{v}}} \end{pmatrix}\right|^2 = \left|\begin{pmatrix} v_1 \cos(\theta)e^{i\phi_1} - v_2 \sin(\theta)e^{i(\phi_2+\theta_{\boldsymbol{v}})} \\ v_1 \sin(\theta)e^{i\phi_3} + v_2 \cos(\theta)e^{i(\phi_2+\phi_3-\phi_1+\theta_{\boldsymbol{v}})} \end{pmatrix}\right|^2 = \left|\begin{pmatrix} v_1 \cos(\theta) - v_2 \sin(\theta)e^{i(\phi+\theta_{\boldsymbol{v}})} \\ v_1 \sin(\theta) + v_2 \cos(\theta)e^{i(\phi+\theta_{\boldsymbol{v}})} \end{pmatrix}\right|^2$$
avec $\phi = \phi_2 - \phi_1$. La matrice $\mathbf{E}_k$ peut être remplacée par une matrice qui s'écrit avec 2 paramètres non redondants :
$$\mathbf{E}_k(\theta, \phi) = \begin{pmatrix} \cos(\theta) & -\sin(\theta)e^{i\phi} \\ \sin(\theta) & \cos(\theta)e^{i\phi} \end{pmatrix}. \tag{2.1}$$

En pratique, on n'utilisera que les trois types de mesures suivant :

$$\mathbf{E}_X = \mathbf{E}_k(\tfrac{\pi}{4}, \pi) = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{E}_Y = \mathbf{E}_k(\tfrac{\pi}{4}, \tfrac{\pi}{2}) = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}, \quad \mathbf{E}_Z = \mathbf{E}_k(0, 0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.2}$$

À un renommage des résultats de mesure près, ce sont les trois mesures de Pauli non triviales sur un qubit. Les mesures de Pauli sont les mesures projectives définies par les matrices hermitiennes ($\mathbf{H}_{\mathcal{M}}$, voir section 1.1.3) de Pauli non triviales :

$$\boldsymbol{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \boldsymbol{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Les transconjuguées des lignes des matrices $\mathbf{E}_X, \mathbf{E}_Y, \mathbf{E}_Z$ sont des vecteurs propres des matrices $\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z$; et les valeurs propres associées $+1$ pour la première ligne et -1 pour la deuxième ligne. C'est ici qu'est la seule différence entre nos mesures et les mesures de Pauli. Les résultats possibles des mesures que nous considérons sont 0 et 1 et pas -1 et 1. Cependant, ce sont les probabilités de chaque résultat (caractérisées par les vecteurs propres) qui nous intéressent, et pas les valeurs que la mesure peut prendre (les valeurs propres). On dit donc que les mesures de Pauli et nos mesures sont équivalentes.

La quatrième matrice de Pauli est l'identité, la mesure quantique associée n'a pas d'intérêt, car elle n'a qu'une seule valeur propre : $+1$, donc un seul résultat possible. Les trois mesures de Pauli ont une grande importance dans la physique quantique. En effet, si le qubit est réalisé par le spin d'un électron, alors, avec les conventions classiques, mesurer la composante du spin dans les trois directions de l'espace X, Y, Z correspond aux trois mesures que nous avons définies. La seule différence est que les résultats possibles ne sont pas les mêmes, on fait correspondre une composante de spin (suivant un axe quelconque) égale à $1/2$ en unité normalisée à la valeur 0 pour nos mesures (ou 1 pour les mesures de Pauli) et une composante de spin de $\frac{-1}{2}$ à la valeur 1 pour nos mesures (ou -1 pour les mesures de Pauli).

Si le qubit est réalisé par autre chose que le spin d'un électron, alors on a toujours un équivalent à la mesure selon Z (il y a toujours une mesure dans la base de référence), mais les mesures selon X et Y peuvent être plus difficiles à réaliser expérimentalement. Pour certaines architectures (celles qui sont testées dans [CFYW19] par exemple), seules les mesures dans la base de référence sont possibles, et si on veut faire les mesures associées à $\mathbf{E}_X$ et $\mathbf{E}_Y$, il faut utiliser des portes quantiques mono-qubit :

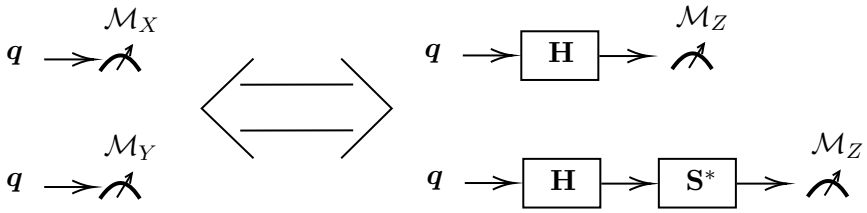


FIGURE 2.2 : Réalisation des mesures $\mathcal{M}_X$ et $\mathcal{M}_Y$ pour les architectures de qubits qui ne permettent que des mesures dans la base de référence. $\mathbf{H}$ est la porte de Hadamard, et $\mathbf{S}^*$ représente la porte phase définie par la matrice $\begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$.

Que les mesures quantiques soient effectuées directement (schéma de gauche sur la figure 2.2) ou avec des portes quantiques (schéma de droite), il peut y avoir des erreurs. C'est-à-dire que les matrices $\mathbf{E}_X$ et $\mathbf{E}_Y$ peuvent ne pas correspondre à celles du modèle (2.2) soit parce que les directions dans lesquelles on mesure la composante de spin (ou une autre quantité représentée par le qubit) ne sont pas tout à fait bonnes, soit parce que les portes quantiques mono qubit sont mal réalisées. Nous verrons dans le chapitre 4 comment on peut corriger (dans une certaine mesure) ce type d'erreur. Dans le reste de la thèse, nous considérons que ces mesures sont exactes et elles servent de référence, c'est-à-dire que $|0\rangle$ et $|1\rangle$ sur chaque qubit sont définis (à une phase près) comme les états qui, quand ils sont mesurés avec $\mathcal{M}_Z$, donnent 0 et 1 respectivement avec

probabilité 1. L'écart de phase entre $|0\rangle$ et $|1\rangle$ est fixé en imposant que $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ donne 0 avec probabilité 1 quand on le mesure avec $\mathcal{M}_X$.

Pour $n_{qb} > 1$ on fait les mesures $\mathcal{M}_X$, $\mathcal{M}_Y$ ou $\mathcal{M}_Z$ simultanément sur chaque qubit comme sur la figure 2.1. En tout, il y a $3^{n_{qb}}$ mesures possibles, mais nous ne les utiliserons pas toutes, nous n'utiliserons que $n_t = 4$ ou $n_t = 15$ types de mesures. Chaque type de mesure sera réalisé $n_c > 1$ fois pour estimer les probabilités d'occurrence, nous ferons les tests avec $n_c n_t = 5000$ et $n_c n_t = 500000$. $n_c n_t$ est important, car c'est le nombre total de mesures qui sont réalisés sur l'état à identifier.

On peut montrer (voir A.2 annexe) que les matrices de vecteurs propres associées aux mesures multi-qubit sont les produits tensoriels des matrices de vecteurs propres 2×2 des mesures que l'on fait sur chaque qubit. Par exemple, si on mesure le premier qubit avec $\mathcal{M}_Z$ et le deuxième avec $\mathcal{M}_X$, la mesure que l'on obtient est $\mathcal{M}_{ZX}$ et sa matrice de vecteurs propres est :

$$\mathbf{E}_{ZX} = \mathbf{E}_Z \otimes \mathbf{E}_X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Ce choix de mesures projectives non intriquées et à d résultats possibles est présent dans la littérature relative au traitement de l'information quantique en général, surtout dans les domaines proches de l'implémentation physique ([XCZ+18] [WBC+21] [DDH+22] [CFYW19]). Cependant, dans la littérature sur la QST et QPT, il est très rare de voir des articles qui considèrent ces mesures pour plus d'un qubit (c'est le cas dans [CW20] [SSKKG22]), et certains articles (comme [GCE+15]) reconnaissent que les mesures non intriquées sont plus simples à réaliser que les mesures intriquées.

2.1.2 Inconvénients des mesures de Pauli multi-qubit

La présente section (ainsi que la suivante, section 2.1.3) n'est pas nécessaire pour comprendre le reste du manuscrit, elle présente les autres types de mesures utilisés dans la littérature et explique pourquoi elles ne nous conviennent pas. Cela explique pourquoi nous avons dû développer nos algorithmes originaux pour la QST plutôt que d'utiliser ceux de la littérature.

Les mesures non intriquées les plus présentes dans la littérature sur la QST sont les mesures de Pauli. Elles sont équivalentes aux mesures que nous considérons seulement dans le cas $n_{qb} = 1$. Dans le cas général, il y a $4^{n_{qb}}$ mesures de Pauli, les matrices hermitiennes qui définissent ces mesures sont les produits tensoriels de n_{qb} matrices de Pauli $\{\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z, \boldsymbol{\sigma}_I = \mathbf{I}_2\}$. Par exemple, la mesure de Pauli ZX est définie par la matrice hermitienne suivante :

$$\boldsymbol{\sigma}_{ZX} = \boldsymbol{\sigma}_Z \otimes \boldsymbol{\sigma}_X = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

On peut montrer que toutes ces matrices n'ont que deux valeurs propres distinctes : $+1$ et -1 et que les espaces propres associés sont de dimension $\frac{d}{2}$. Donc, si l'on considère un système qui réalise physiquement une mesure de Pauli sur un état donné, ce système est hautement sous optimal. Présentons un exemple qui illustre pourquoi : si l'état que l'on mesure est totalement inconnu, et que l'on donne l'a priori que le vecteur d'état est la réalisation (on réechantillonne la densité pour chaque répétition de l'expérience) d'un vecteur aléatoire dont la distribution est uniforme sur l'espace de Hilbert (i.e. il s'agit de l'état mélange représenté par $\boldsymbol{\rho} = \frac{1}{d}\mathbf{I}_d$), alors, quelle que soit la mesure que l'on effectue sur cet état, tous les résultats possibles ont la même probabilité d'occurrence : $\frac{d_k}{d}$ où d_k est la dimension de l'espace propre associé au résultat (car cette probabilité est la trace de $\frac{1}{d}\mathbf{I}_d$ multipliée par la matrice de projection sur l'espace propre, voir section 1.1.8.3). Donc pour une mesure de Pauli, on obtient -1 avec une probabilité $\frac{1}{2}$,

et $+1$ avec une probabilité $\frac{1}{2}$. Or pour nos types de mesures à d résultats possibles, chacun des résultats possibles a un espace propre associé de dimension 1 (par définition, les vecteurs directeurs sont les lignes de la matrice des vecteurs propres), donc tous les résultats possibles sont équiprobables de probabilité $\frac{1}{d}$. L'information (classique) de Shannon apportée par notre mesure est $-\lg(\frac{1}{d}) = n_{qb}$ bits d'information et c'est l'information maximale qu'une mesure à d résultats possibles peut apporter. L'information (classique) de Shannon apportée par une mesure de Pauli est $-\lg(\frac{1}{2}) = 1$ bit d'information, ce qui est certes la quantité d'information maximale qu'une mesure à 2 résultats possibles peut apporter, mais correspond à moins d'information moins que n_{qb} bits. C'est en ce sens que nous pensons que nos mesures sont plus pertinentes que les mesures de Pauli, ces deux mesures ont le même coût (l'état mesuré est perturbé et ne sera plus utilisé), mais, en général, une mesure à d résultats possibles apporte beaucoup plus d'information sur le système qu'une mesure à 2 résultats possibles.

Les mesures de Pauli ont deux avantages cependant :

1. Les matrices de Pauli forment une base orthogonale de l'espace vectoriel des matrices hermitiennes sur le corps des réels muni du produit scalaire de Hilbert-Schmidt $\langle \mathbf{A}, \mathbf{B} \rangle = Re\,(tr(\mathbf{A}\mathbf{B}^*)) = tr(\mathbf{A}\mathbf{B})$ pour des matrices hermitiennes

2. À partir de n_c réalisations de chacune des $3^{n_{qb}}$ mesures que nous avons définies dans la partie précédente sur un même état, on peut avoir $3^{n_i} n_c$ réalisations synthétiques de chacune des $4^{n_{qb}}$ mesures de Pauli sur le même état, n_i étant "nombre de I" dans la chaine de caractère de la mesure de Pauli que l'on considère (par exemple pour la mesure de Pauli zx (représentée par σ_{zx}) $n_i = 0$ et pour la mesure de Pauli zi, $n_i = 1$). Cette méthode est très intéressante car (i) les mesures qui sont effectuées en pratique sont bien non-intriquées et ont bien d résultats possibles, en plus de ne pas être redondantes, (ii) on ne perd pas d'information en passant des $3^{n_{qb}}$ mesures à d résultats possibles aux $4^{n_{qb}}$ mesures de Pauli.

Le premier avantage est facile à montrer : on peut montrer par récurrence sur le nombre de qubits que les matrices de Pauli forment une base orthogonale et que les normes de chaque matrice sont $\sqrt{d}$. L'espérance de la mesure de Pauli définie par $\boldsymbol{\sigma}_s$ (s est une chaîne de caractères composée de x, de y, de z et de i) sur l'état représenté par $\boldsymbol{\rho}$ est $tr(\boldsymbol{\sigma}_s \boldsymbol{\rho})$ (voir section 1.1.8.2). Donc, l'espérance de la mesure de Pauli est le produit scalaire de la matrice de Pauli avec la matrice densité. Il suffit de diviser par le carré de la norme de la matrice de Pauli (d) pour obtenir la décomposition de $\boldsymbol{\rho}$ dans la base des matrices de Pauli :

$$\boldsymbol{\rho} = \sum_{s \in \{X,Y,Z,I\}^{n_{qb}}} \frac{tr(\boldsymbol{\sigma}_s \boldsymbol{\rho})}{d} \boldsymbol{\sigma}_s \tag{2.3}$$

Cette équation est la même que l'équation (8.149) de [NC00], où les auteurs proposent (implicitement) d'utiliser les mesures de Pauli (sans les nommer) pour réaliser de la tomographie d'état, mais ne donnent pas de détails sur la façon dont les mesures de Pauli peuvent être réalisées matériellement.

Le deuxième avantage est très important pour réaliser des mesures de Pauli et est très souvent ignoré dans la littérature (nous n'avons trouvé l'idée que dans [CW20]). Nous partons du constat que, par construction, le résultat de la mesure de Pauli s définie par $\sigma_s = \sigma_{c_1} \otimes ... \otimes \sigma_{c_{n_{qb}}}$ avec $s = c_1 ... c_{n_{qb}}$ (où chaque $c_k \in \{x, y, z, i\}$ est un caractère, et s est la chaîne des n_{qb} caractères) est le produit des n_{qb} résultats des mesures de Pauli s_k sur le k-ème qubit. Nos mesures font presque la même chose, mais au lieu de faire le produit des résultats (ce qui fait perdre beaucoup d'information), on choisit de renommer les -1 en 1, renommer les $+1$ en 0 et de les concaténer pour avoir une chaîne de bits. Avec une réalisation de l'un de nos types de mesures $\mathcal{M}_{s^0}$, on peut donc générer une réalisation de chacune des $2^{n_{qb}}$ mesures de Pauli définies par les matrices

hermitiennes $\boldsymbol{\sigma}_{s_1}, ..., \boldsymbol{\sigma}_{s_d}$ où le k-ème caractère de la chaîne de caractères s_j est le k-ème caractère de s_0 si le k ème bit de $j-1$ en base 2 est 0 et est i sinon. Par exemple, pour $n_{qb} = 2$, si $s_0 = ZX$, on effectue une mesure de $\mathcal{M}_{ZX}$, si on obtient e.g. 01, alors on sait que la mesure de Pauli Z sur le premier qubit vaut $+1$ et la mesure de Pauli X sur le deuxième qubit vaut -1. On sait aussi que la mesure de Pauli I vaut toujours $+1$ par définition. On peut donc générer une réalisation des mesures de Pauli définies par $\boldsymbol{\sigma}_{ZX}, \boldsymbol{\sigma}_{ZI}, \boldsymbol{\sigma}_{IX}$ et $\boldsymbol{\sigma}_{II}$, ces réalisations seraient respectivement $+1 \times -1 = -1$, $+1 \times +1 = +1$, $+1 \times -1 = -1$ et $+1 \times +1 = +1$. Si, au lieu d'avoir une réalisation d'un seul type de mesure $\mathcal{M}_{s_0}$, on a n_c réalisations de chacune des $3^{n_{qb}}$ mesures de $\{\mathcal{M}_s\}_{s\in\{X,Y,Z\}^{n_{qb}}}$, alors, on peut obtenir des réalisations de toutes les mesures de Pauli, le nombre de réalisations dépend du "nombre de I" dans la mesure que l'on considère. Si $n_{qb} = 3$ et que l'on veut avoir des réalisations de la mesure de Pauli IIX par exemple, on peut utiliser les réalisations des mesures $\mathcal{M}_{XXX}, \mathcal{M}_{XYX}, \mathcal{M}_{XZX}, \mathcal{M}_{YXX}, \mathcal{M}_{YYX}, \mathcal{M}_{YZX}, \mathcal{M}_{ZXX}, \mathcal{M}_{ZYX}$ et $\mathcal{M}_{ZZX}$, on peut donc avoir $9n_c$ réalisations de la mesure de Pauli IIX. Dans le cas général, on peut avoir $3^{n_i} n_c$ réalisations de chaque mesure de Pauli.

En tirant parti des deux avantages que nous avons listés, il est donc possible de calculer des réalisations de toutes les mesures de Pauli de façon "synthétique" et efficace (avec les mesures à d résultats possibles que nous avons définies), et de s'en servir pour obtenir la décomposition de la matrice densité $\boldsymbol{\rho}$ de l'état que l'on mesure. Cependant, aucun article de QST qui utilise des matrices de Pauli ne précise comment elles doivent être réalisées en pratique (sauf [CW20] qui décrit rapidement la méthode dont nous venons de parler sans rentrer autant dans les détails). Et certains articles [KKD15], [SRA+13], [GLF+10], [MJZ+16], [CKW+16] et [Wan13] proposent d'utiliser seulement des sous-ensembles de mesure de Pauli (pas toujours bien définis, [KKD15] propose de choisir des mesures de Pauli aléatoirement par exemple) en ignorant le fait que les mesures de Pauli ne sont pas effectuées indépendamment mesure par mesure. Dans [MJZ+16] par exemple, les auteurs proposent d'utiliser les mesures de Pauli $\{II, IX, IY, IZ, XI, YX, YY, YZ, ZX, ZY, ZZ\}$ pour deux qubits, ils montrent que tous les états purs peuvent être identifiés (parmi les états purs) de façon unique avec les espérances de mesures de Pauli sur l'état (ils doivent utiliser la mesure de Pauli II qui, normalement, est inutile, parce qu'ils relaxent la contrainte qui impose à la trace de la matrice densité d'être unitaire, pour des raisons assez floues). Pour nous, l'intérêt de cet article est purement théorique, en pratique, si l'on veut effectuer les 11 mesures de Pauli que les auteurs proposent, il faut (à notre connaissance) passer par $\{\mathcal{M}_{YX}\, \mathcal{M}_{YY}\, \mathcal{M}_{YZ}\, \mathcal{M}_{ZX}\, \mathcal{M}_{ZY}\, \mathcal{M}_{ZZ}\}$ (pour avoir toutes les mesures de Pauli sauf XI) et une mesure comme $\mathcal{M}_{XZ}$ qui mesure le premier qubit selon X pour avoir XI (nous aurions aussi pu choisir $\mathcal{M}_{XY}$ ou $\mathcal{M}_{XZ}$). Il existe peut-être d'autres méthodes, mais, si c'est le cas, nous avons des doutes sur leur optimalité. En effet, les mesures de Pauli multi qbit sont définies comme les produits des mesures de Pauli mono-qubit (i.e. mesures de spin pour un électron), et si l'on veut calculer ces produits, nous pensons qu'il est optimal de calculer les mesures de Pauli de chaque qubit simultanément (i.e. calculer les composantes des spins dans les bonnes directions si l'on a affaire à des électrons) puis calculer le produit. Or, si l'on effectue bien les mesures $\mathcal{M}_{YX}\, \mathcal{M}_{YY}\, \mathcal{M}_{YZ}\, \mathcal{M}_{ZX}\, \mathcal{M}_{ZY}\, \mathcal{M}_{ZZ}$ et $\mathcal{M}_{XZ}$, alors, en plus d'avoir les mesures de Pauli dont les auteurs ont besoin $\{II, IX, IY, IZ, XI, YX, YY, YZ, ZX, ZY, ZZ\}$, on a aussi les mesures de Pauli YI (grâce à $\mathcal{M}_{YX}\, \mathcal{M}_{YY}\, \mathcal{M}_{YZ}$) et ZI (grâce à $\mathcal{M}_{ZX}$, $\mathcal{M}_{ZY}$ et $\mathcal{M}_{ZZ}$). En fait, on a donc toutes les mesures de Pauli sauf $\{XX, XY\}$ (il nous faudrait $\mathcal{M}_{XX}$ et $\mathcal{M}_{XY}$).

Pour récapituler ce que nous avons vu sur les mesures de Pauli, nous pensons qu'elles sont souvent mal utilisées dans la littérature et que l'on ne doit pas considérer qu'elles sont réalisées directement. Elles ont l'avantage de rendre la tomographie d'état mélange avec des matrices densité particulièrement simple (voir (2.3)). Mais, comme nous n'utilisons pas les opérateurs densité, nous choisissons d'ignorer les mesures de Pauli bien qu'elles puissent être facilement reconstituées à partir des mesures que nous considérons.

2.1.3 Inconvénients des mesures intriquées et des mesures qui n'ont pas d résultats possibles

Les mesures de Pauli ne sont pas les seuls types de mesures utilisés dans la littérature, mais, si on veut utiliser des algorithmes qui sont adaptés aux états purs ([GCE+15], [FSC05], [Fin04]) ou l'algorithme de SGQT qui nécessite très peu de mesures, alors il faut être capable de réaliser au moins un des circuits de la figure 2.3 :

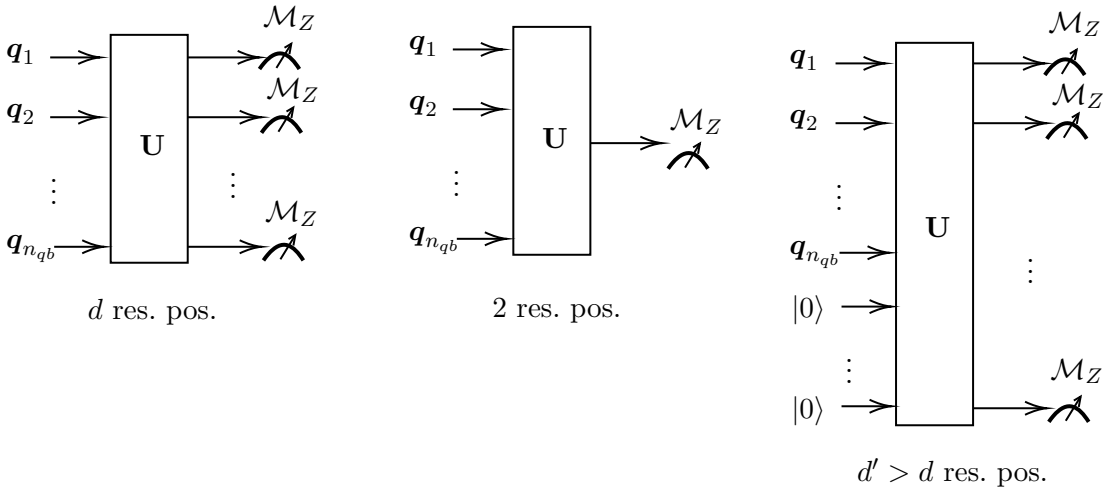


FIGURE 2.3 : Autres types de mesures considérés dans la littérature, de gauche à droite : mesure projective intriquée à d résultats possibles, mesure projective intriquée à 2 résultats possibles, POVM réalisée avec des qubits ancillaires.

- [Fer14] utilise des mesures projectives à deux résultats possibles (circuit du milieu sur la figure 2.3) sur des états propres intriqués non orthogonaux. Chaque itération de l'algorithme calcule un nouveau type de mesure qui doit être réalisée (ce denier dépend de ce qui a été mesuré auparavant et sera très probablement intriqué) et, par conséquent, une nouvelle porte quantique doit être construite à la volée.

- [Fin04] considère des mesures à deux résultats possibles. La moitié de ces mesures peuvent être déduites de $\mathcal{M}_{Z\ldots Z}$, mais l'autre moitié doit être réalisée avec des circuits du type du circuit du milieu sur la figure 2.3.

- [GCE+15] utilise 3 types de mesures non intriquées à d résultats possibles (spécifiquement $\mathcal{M}_{Z\ldots Z}$, $\mathcal{M}_{Z\ldots ZX}$ et $\mathcal{M}_{Z\ldots ZX}$), et deux types de mesures intriquées à d résultats possibles (circuit de gauche de la figure 2.3) pour identifier des états purs. Sur les 5 types de mesures, le premier ($\mathcal{M}_{Z\ldots Z}$) est facultatif (ne sert qu'à vérifier que l'état est bien pur). Les auteurs précisent comment les deux types de mesures intriquées peuvent être réalisés en pratique, et si on veut utiliser le circuit de gauche de la figure 2.3, alors la seule porte multi-qubit dont on a besoin est la porte qui réalise la transformée de Fourier quantique (appliquée 2 fois) [NC00].

- [FSC05] propose une méthode pour réaliser la QST d'un état pur avec un seul type de mesure. Avec un type de mesure à d résultats possibles ou moins, ce serait impossible (l'état pur a trop de paramètres). Les auteurs proposent donc un POVM à $2d$ résultats possibles qui doit être réalisé avec un circuit du type du circuit de droite sur la figure 2.3.

De manière générale, dans la littérature sur traitement de l'information quantique, les auteurs ne se soucient pas vraiment de l'implémentation physique des algorithmes qu'ils proposent, c'est particulièrement apparent pour les mesures. Dans [NC00], les auteurs sont transparents sur ce sujet : Dans la Box 2.5, quand ils expliquent pourquoi ils introduisent les POVM alors que ceux-ci sont généralement ignorés par les physiciens, ils écrivent "The reason most physicists don't learn the general measurement formalism is because most physical systems can only be measured in a very coarse manner. In quantum computation and quantum information we aim for an exquisite level of control over the measurements that may be done, and consequently it helps to use a more comprehensive formalism for the description of measurements.". En effet, le champ du traitement de l'information quantique a été développé sur le papier bien avant que les réalisations matérielles soient prêtes, et les auteurs étaient souvent incités à se concentrer sur le formalisme mathématique, en espérant que les progrès sur les réalisations matérielles permettent un jour à leurs algorithmes d'être réalisés physiquement. Le seul algorithme de tomographie d'état pur que nous avons envisagé d'utiliser est celui de Goyeneche et al. [GCE+15], à nos yeux, il s'agit du plus réaliste. Cependant, même pour [GCE+15] on a besoin d'une porte multi-qubit d'intrication (la transformée de Fourier "au carré") et comme l'objectif principal de la thèse est d'identifier les portes quantiques, nous avons préféré développer nos propres algorithmes de QST d'états purs.

2.2 Tomographie d'états purs en dimension quelconque avec $n_t = 4$ types de mesures

La présente section décrit notre première implémentation de QST, la section 2.2.1 décrit les quatre types de mesures parallèles non intriquées qui sont effectués, la section 2.2.2 explique pourquoi il est raisonnable de penser qu'elles sont injectives à une phase globale près et la section 2.2.3 décrit un premier algorithme qui estime l'état à partir des mesures.

2.2.1 Types de mesures

Dans la méthode de QST décrite ici, nous effectuons quatre types de mesures sur l'état considéré : Le premier mesure tous les qubits selon Z, sa matrice de vecteurs propres, $\mathbf{E}_{Z...Z}$ est la matrice identité, le deuxième mesure tous les qubits selon Y, le troisième selon X, et le quatrième mesure tous les qubits impairs (premier qubit, troisième qubit etc...) selon X et tous les qubits pair selon Y.

Après avoir effectué les mesures plusieurs fois sur des copies de l'état, nous calculons les probabilités empiriques $\widehat{\boldsymbol{p}}_{\mathcal{M}}$ pour $\mathcal{M}$ couvrant les quatre types de mesures. Nous disposons alors d'un vecteur de dimension $n_s = 4d$ avec $\underline{n_s} = 4(d-1)$ degrés de liberté[1] que nous appelons $\widehat{\boldsymbol{p}}_s$. Le vecteur de probabilités théorique associé est $\boldsymbol{p}_s = |\mathbf{A}_s \boldsymbol{v}|^2$ où s signifie "small", car la matrice correspondante que nous introduirons dans la section 2.3 a plus de lignes. $\mathbf{A}_s$ est la concaténation des matrices des vecteurs propres des mesures que nous effectuons, $\underline{\mathbf{A}}_s$ est la version non-redondante de $\mathbf{A}_s$:

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{E}_{Z...Z} \\ \mathbf{E}_{Y...Y} \\ \mathbf{E}_{X...X} \\ \mathbf{E}_{XYXY...} \end{bmatrix} \text{ et } \underline{\mathbf{A}}_s = \begin{bmatrix} \underline{\mathbf{E}}_{Z...Z} \\ \underline{\mathbf{E}}_{Y...Y} \\ \underline{\mathbf{E}}_{X...X} \\ \underline{\mathbf{E}}_{XYXY...} \end{bmatrix} \tag{2.4}$$

où les matrices de vecteurs propres non redondantes $\underline{\mathbf{E}}_{\mathcal{M}}$ sont les matrices de vecteurs propres $\mathbf{E}_{\mathcal{M}}$ avec la dernière ligne enlevée ($\mathcal{M} \in \{Z...Z, Y...Y, X...X, XY...\}$). On dit que ces matrices

[1] car la somme des probabilités empiriques pour un type de mesure donnée vaut à 1

sont non redondantes car, comme on sait que $\boldsymbol{v}$ est de norme 1, on sait que les probabilités théoriques contenues dans le vecteur $|\mathbf{E}_{\mathcal{M}}\boldsymbol{v}|^2$ somment à 1 (quel que soit le type de mesure $\mathcal{M}$), elles sont donc redondantes (on peut enlever une des probabilités sans perdre d'information). Si on enlève le dernier élément du vecteur $|\mathbf{E}_{\mathcal{M}}\boldsymbol{v}|^2$ (ou la dernière ligne de $\mathbf{E}_{\mathcal{M}}$) on a $|\underline{\mathbf{E}}_{\mathcal{M}}\boldsymbol{v}|^2$ qui n'est plus redondant (dans le sens où si on ne connait pas $\boldsymbol{v}$, alors enlever un élément du vecteur nous fait perdre de l'information). On peut retrouver $|\mathbf{E}_{\mathcal{M}}\boldsymbol{v}|^2$ à partir de $|\underline{\mathbf{E}}_{\mathcal{M}}\boldsymbol{v}|^2$, il suffit de soustraire la somme des éléments à 1 et de placer le résultat à la fin du vecteur. Les probabilités non redondantes sont $\underline{\boldsymbol{p}_s} = |\underline{\mathbf{A}_s}\boldsymbol{v}|^2$. $\underline{\boldsymbol{p}_s}$ et $\boldsymbol{p}_s$ contiennent la même information (les éléments enlevés dans $\underline{\boldsymbol{p}_s}$ valent valent 1 moins la somme des $d-1$ éléments précédents).

Dans la section 2.2.2, nous utilisons $\underline{\mathbf{A}_s}$, $\underline{n_s}$ et $\underline{\boldsymbol{p}_s}$ afin de voir si les mesures sont injectives, car nous ne voulons pas introduire de redondance lors du comptage des mesures. Mais, par souci de simplicité, nous utilisons $\mathbf{A}_s$, n_s et $\boldsymbol{p}_s$ dans la section 2.2.3 afin de récupérer l'état à partir des mesures. Nous voulons utiliser toutes les mesures de $\widehat{\boldsymbol{p}_s}$ qu'elles soient redondantes ou non.

2.2.2 Injectivité

$\underline{\mathbf{A}_s}$ est une matrice $\underline{n_s} \times d$ connue et $\boldsymbol{v}$ un vecteur inconnu de norme unitaire. Nous voulons savoir si les mesures que nous avons choisies sont suffisantes pour récupérer n'importe quel $\boldsymbol{v}$ à partir de $|\underline{\mathbf{A}_s}\boldsymbol{v}|^2$ à une phase globale près. Dans la suite du document, cette propriété sera appelée injectivité. C'est un peu exagéré car $\boldsymbol{v} \to |\underline{\mathbf{A}_s}\boldsymbol{v}|^2$ n'est jamais vraiment injective, car changer la phase globale de $\boldsymbol{v}$ ne change pas $|\underline{\mathbf{A}_s}\boldsymbol{v}|^2$. Cette question de l'injectivité a déjà été étudiée dans [HMW13], [BCE06], [BCMN14] dans un cadre légèrement différent : les mesures considérées sont $|\underline{\mathbf{A}_s}\boldsymbol{v}|$ au lieu de $|\underline{\mathbf{A}_s}\boldsymbol{v}|^2$ mais cela ne change rien à l'injectivité. En outre, $\boldsymbol{v}$ n'est pas supposé avoir une norme unitaire, ce qui est important. Afin de réconcilier les deux configurations, nous pouvons relâcher l'hypothèse de norme unitaire pour $\boldsymbol{v}$ et insérer la ligne $[0, ..., 0, 1]$ entre la $(d-1)$-ème et la ligne d-ème ligne (nous aurions pu choisir une autre position) de $\underline{\mathbf{A}_s}$. Cela garantit que la norme de $\boldsymbol{v}$ est contrainte : son carré est la somme des d premières mesures, car les d premières lignes $\underline{\mathbf{A}_s}$ sont la matrice d'identité. Avec ce changement, $\underline{\mathbf{A}_s}$ a $4d-3$ lignes.

Selon [HMW13], le nombre minimal de lignes pour $\underline{\mathbf{A}_s}$ en dessous duquel l'injectivité est impossible est de $4d-3-c(d)n_{qb}$ lignes pour un certain $c(d) \in [1, 2]$. Puisque nous avons $4d-3$ lignes, cette condition nécessaire est satisfaite. Cependant, il n'existe pas de condition suffisante simple sur $\underline{\mathbf{A}_s}$ qui assure l'injectivité. Et prouver l'injectivité pour une matrice $\underline{\mathbf{A}_s}$ donnée est un problème difficile connu. Le résultat le plus proche d'une condition suffisante que nous ayons trouvé dans la littérature est dans [BCE06] où l'on montre que pour une matrice $\underline{\mathbf{A}_s}$ générique, le fait d'avoir $4d-2$ lignes ou plus assure l'injectivité. $\underline{\mathbf{A}_s}$ doit être générique dans le sens où elle fait partie d'un ensemble spécifique ouvert dense de mesure pleine. Nous ne pouvons pas identifier cet ensemble et vérifier que $\underline{\mathbf{A}_s}$ s'y trouverait (bien qu'il s'y trouverait probablement puisque l'ensemble est de mesure pleine), mais cela n'aurait pas suffi à montrer l'injectivité, car il nous manque une ligne pour satisfaire la condition de [BCE06]. Cependant, [BCMN14] a expliqué pourquoi il est naturel de penser que $4d-4$ est la vraie borne inférieure. Il convient de noter que cela reste une conjecture.

Nous pouvons être sûrs que trois types de mesures ne seraient pas suffisants pour atteindre l'injectivité avec $n_{qb} > 2$ car la limite de [HMW13] ne serait pas respectée : nous aurions $3d-2$ lignes indépendantes ($3d-3$ plus la contrainte de norme unitaire). Ce nombre est toujours strictement inférieur à $4d-3-2n_{qb}$ pour $n_{qb} > 2$. $n_t = 4$ est le plus petit nombre de types de mesures pour lequel nous pouvons toujours espérer obtenir l'injectivité.

En résumé, nous ne sommes pas en mesure de prouver l'injectivité pour les mesures définies par (2.4), et la validité d'un algorithme QST associé ne sera testée qu'avec les simulations de la section 2.5. Nous avons choisi d'utiliser quatre types de mesures pour que l'injectivité

soit techniquement possible (et probable). Les types de mesures que nous avons choisis sont arbitraires (bien que nous ayons veillé à sélectionner des matrices de vecteurs propres bien différentes pour éviter les problèmes de mauvais conditionnement).

2.2.3 Une première méthode de QST

Dans la présente section, nous montrons comment la méthode proposée dans [WdM13] peut être utilisée dans notre cadre pour récupérer $\boldsymbol{v}$ à partir des probabilités empiriques $\widehat{\boldsymbol{p}_s}$. Le vecteur $\widehat{\boldsymbol{p}_s}$ est une estimation de $\boldsymbol{p}_s = |\mathbf{A}_s\boldsymbol{v}|^2$ (nous ne considérons que $\mathbf{A}_s$ à partir de maintenant, $\underline{\mathbf{A}_s}$ n'était utile que pour étudier l'injectivité). Le problème d'optimisation considéré dans [WdM13] est le suivant :

$$\min_{\boldsymbol{v}} \left\| |\mathbf{A}_s\boldsymbol{v}| - \sqrt{\widehat{\boldsymbol{p}_s}} \right\| \tag{2.5}$$

où $\sqrt{\widehat{\boldsymbol{p}_s}}$ est la racine carrée élément par élément de $\widehat{\boldsymbol{p}_s}$ et $||.||$ est la norme L_2. La contrainte $||\boldsymbol{v}||^2 = 1$ est implicite dans le critère à minimiser. En effet, la somme des d premiers éléments de $|\mathbf{A}_s\boldsymbol{v}|^2$ est la norme quadratique de $\boldsymbol{v}$ et la somme des d premiers éléments de $\widehat{\boldsymbol{p}_s}$ est 1, donc si $|\mathbf{A}_s\boldsymbol{v}|$ est proche de $\sqrt{\widehat{\boldsymbol{p}_s}}$ leurs normes au carré seront également proches, et donc la norme au carré de $\boldsymbol{v}$ sera proche de 1. Dans [WdM13], il est montré que (2.5) est équivalent au problème d'optimisation suivant (à l'origine, la preuve vient de [Sho87]) :

$$\min_{\mathbf{U} \text{ t.q. } \mathcal{C}} tr(\mathbf{US}) \tag{2.6}$$

où $\mathbf{S} = diag(\sqrt{\widehat{\boldsymbol{p}_s}})(I - \mathbf{A}_s\mathbf{A}_s^\dagger)diag(\sqrt{\widehat{\boldsymbol{p}_s}})$, † est la pseudo-inverse, $diag(\sqrt{\widehat{\boldsymbol{p}_s}})$ est la matrice diagonale dont la diagonale est $\sqrt{\widehat{\boldsymbol{p}_s}}$ et $\mathcal{C}$ représente la condition suivante sur la matrice $n_s \times n_s$ notée $\mathbf{U}$:

$$\exists \boldsymbol{u} \in \mathbb{C}^{n_s} \text{ t.q. } |\boldsymbol{u}| = [1, ..., 1]^T \text{ et } \mathbf{U} = \boldsymbol{u}\boldsymbol{u}^*. \tag{2.7}$$

[WdM13] montre que si $\mathbf{U}$ est une solution de (2.6), alors, le vecteur $\boldsymbol{u}$ associé de (2.7) est une approximation de la phase de $\mathbf{A}_s\boldsymbol{v}$, et l'estimée de $\boldsymbol{v}$ qui en découle est définie comme :

$$\widehat{\boldsymbol{v}}_0 = \mathbf{A}_s^\dagger(\boldsymbol{u} \odot \sqrt{\widehat{\boldsymbol{p}_s}}) \tag{2.8}$$

($\odot$ est le produit élément par élément) est la solution de (2.5) proposée dans [WdM13].

Nous ne détaillons pas la preuve ici, mais l'intuition derrière la formulation de (2.6) est assez simple : le critère de (2.6) peut être réécrit comme $(\boldsymbol{u} \odot \sqrt{\widehat{\boldsymbol{p}_s}})^*(I - \mathbf{A}_s\mathbf{A}_s^\dagger)(\boldsymbol{u} \odot \sqrt{\widehat{\boldsymbol{p}_s}})$. Le vecteur $\boldsymbol{u} \odot \sqrt{\widehat{\boldsymbol{p}_s}}$ est notre estimation de $\mathbf{A}\boldsymbol{v}$ et $\mathbf{I} - \mathbf{A}_s\mathbf{A}_s^\dagger$ est la projection sur le complément de l'image de $\mathbf{A}$ (le noyau de $\mathbf{A}^*$). Par conséquent, nous recherchons les phases $(\boldsymbol{u})$ qui rapprochent le plus possible notre estimation de $\mathbf{A}\boldsymbol{v}$ de l'image de $\mathbf{A}$ (plus précisément, elles minimisent la norme de la projection sur le noyau de $\mathbf{A}^*$).

(2.6) est presque un problème d'optimisation convexe. En effet, si $\mathcal{C}$ est reformulé de manière équivalente : $\mathbf{U}_{i,i} = 1 \; \forall i \in [1, n_s], \mathbf{U} \succeq 0, \text{rang}(\mathbf{U}) = 1$ ($\mathbf{U} \succeq 0$ signifie que $\mathbf{U}$ est à la fois hermitien et défini positif), alors selon [WdM13] le critère $tr(\mathbf{US})$ est convexe et la seule contrainte qui rend le problème non convexe dans $\mathcal{C}$ est $\text{rang}(\mathbf{U}) = 1$. En la relâchant, on obtient un problème convexe qui peut être résolu sans avoir besoin d'une bonne initialisation

$$\min_{\mathbf{U} \text{ t.q. } \mathbf{U}_{i,i}=1 \; \forall i, \mathbf{U} \succeq 0} tr(\mathbf{US}). \tag{2.9}$$

Sans bruit, la solution que nous recherchons : $\mathbf{U}_0 = \boldsymbol{u}_0\boldsymbol{u}_0^*$ (où $\boldsymbol{u}_0$ est défini comme le vecteur qui contient les facteurs de phase (élément par élément) de $\mathbf{A}\boldsymbol{v}$) est une solution à la fois du problème relaxé (2.9) et du problème original (2.6), car les critères de ces deux problèmes sont positifs ($\mathbf{S}$ est une matrice positive) et il est facile de vérifier que (sans bruit) $tr(\mathbf{U}_0\mathbf{S}) = 0$. Cela

ne garantit cependant pas que la relaxation ne change rien, car le minimum peut ne pas être unique. Comme dans la plus grande partie de [WdM13], nous choisissons d'ignorer ce problème, car comme nous verrons plus loin, la solution du problème relaxé ne sera utilisée que pour initialiser un algorithme d'optimisation non convexe plus rapide et plus précis, nous pouvons donc tolérer de petites erreurs.

Une fois que (2.9) est résolu à l'aide de l'algorithme PhaseCut de [WdM13] (réécrit dans l'annexe A.3), les vecteurs propres et les valeurs propres de la solution $\mathbf{U}$ sont calculés. Afin d'obtenir une estimation de $\boldsymbol{u}$ [WdM13] calcule alors $\widehat{\boldsymbol{u}}$ qui est le vecteur propre associé à la plus grande valeur propre de $\mathbf{U}$. À partir de $\widehat{\boldsymbol{u}}$ nous obtenons l'estimation de $\boldsymbol{v}$ définie dans (2.8) :

$$\widehat{\boldsymbol{v}}_{pc} = \mathbf{A}_s^{\dagger}(\widehat{\boldsymbol{u}} \odot \sqrt{\widehat{\boldsymbol{p}_s}}). \tag{2.10}$$

Dans [WdM13], cette méthode est testée avec des matrices $\mathbf{A}$ qui représentent des cas d'utilisation habituels dans la communauté du traitement des signaux et des images (transformée de Fourier suréchantillonnée, "multiple random illumination filters", transformée en ondelettes) pour lesquels PhaseCut fonctionne bien. Cependant, pour notre matrice $\mathbf{A} = \mathbf{A}_s$, PhaseCut est un bon point de départ mais nécessite ensuite un réglage fin que nous détaillerons dans la section 2.4.

2.2.4 Comparaison avec la littérature

Résumons les principales caractéristiques de notre premier algorithme de QST :

- Il utilise $4d$ probabilités qui peuvent être obtenues en faisant la moyenne des résultats de 4 mesures parallèles non intriquées.
- Il est raisonnable de penser que les mesures choisies sont injectives.
- L'algorithme qui reconstruit l'état n'est pas explicite (optimisation).

La méthode de l'article de Goyeneche et al. [GCE+15] utilise le même nombre de types de mesures, n'est pas injective (mais l'ensemble des états qui ne peuvent pas être identifiés à une phase près est de mesure nulle) et fournit un algorithme de reconstruction explicite. Le principal avantage de notre approche basée sur PhaseCut par rapport à [GCE+15] est que nous n'utilisons pas de mesures intriquées.

Si on se compare à l'approche parcimonieuse plus générale [GLF+10] qui nécessite $O(rd\log(d)^2)$ probabilités pour estimer l'état où r est le rang de la matrice densité vaut 1 dans le cas d'un état pur. Ces probabilités pourraient être obtenues en faisant la moyenne des résultats de $O(\log(d)^2)$ différentes mesures non intriquées, alors, notre méthode est plus efficace puisque nous utilisons $4 = O(1)$ mesures différentes non intriquées. Les deux méthodes n'ont aucune garantie théorique d'injectivité ou de solution explicite (on doit résoudre un problème d'optimisation). La validité des solutions ne peut être démontrée que par des simulations.

2.3 Solution explicite pour la QST

La présente section décrit un autre algorithme de QST qui fonctionne avec des mesures différentes, l'algorithme a l'avantage d'être explicite (pas d'optimisation convexe), mais il a besoin de plus de types de mesures.

2.3.1 Autres types de mesures

Dans la méthode alternative de QST décrite ici, nous effectuons les mesures suivantes :
$\left\{ \underbrace{Z...Z}_{n_{qb} \text{ fois}}, \left\{ \underbrace{Z...Z}_{n_{qb}-i \text{ fois}} \; S \; \underbrace{X...X}_{i-1 \text{ fois}}, 1 \leq i \leq n_{qb} \quad S \in \{X, Y\} \right\} \right\}$.

Le nombre de types de mesures est $2n_{qb}+1$. Le $\mathbf{A}_t$ (t signifie "tall") qui en résulte a $d(2n_{qb}+1)$ lignes, elle est la concaténation des matrices de vecteurs propres ($\mathbf{E}$) associées aux mesures :

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{E}_{Z...Z} \\ \mathbf{E}_{Z...ZX} \\ \mathbf{E}_{Z...ZY} \\ \vdots \\ \mathbf{E}_{X...X} \\ \mathbf{E}_{YX...X} \end{bmatrix} \tag{2.11}$$

Chaque mesure est effectuée plusieurs fois et nous calculons les probabilités empiriques $\widehat{\boldsymbol{p_t}}$ qui sont des estimations des probabilités théoriques $\boldsymbol{p_t} = |\mathbf{A}_t \boldsymbol{v}|^2$.

$2n_{qb}+1$ peut sembler beaucoup comparé aux 4 types de mesures de la section 2.2, mais il s'agit d'une petite fraction des $3^{n_{qb}}$ types de mesures possibles définis dans la section 2.1.1. Cette configuration présente également l'avantage d'offrir un moyen intéressant de récupérer l'état à partir des mesures, comme nous l'expliquerons dans la section 2.3.2.

2.3.2 Algorithme récursif de QST

Montrons comment un vecteur $\boldsymbol{v}$ peut être estimé à une phase globale près à partir de $|\mathbf{A}_t \boldsymbol{v}|^2$ par récurrence sur le nombre de qubits.

$\mathbf{A}_t$ dépend de n_{qb}. Dans le reste de la présente section, cette dépendance ne sera pas omise et $\mathbf{A}_t$ sera appelé $\mathbf{A}_t(n_{qb})$. Nous montrons d'abord comment résoudre le problème (récupérer $\boldsymbol{v}$ à partir de $|\mathbf{A}_t \boldsymbol{v}|^2$) avec $n_{qb} = 1$. Nous expliquons ensuite comment la résolution du problème pour $n_{qb} - 1$ qubits donne la solution pour n_{qb} qubits. À partir de cela, un algorithme récursif peut être implémenté.

Supposons que $n_{qb} = 1$: $\mathbf{A}_t(1) = \begin{bmatrix} \mathbf{E}_Z \\ \mathbf{E}_X \\ \mathbf{E}_Y \end{bmatrix}$, avec les $\mathbf{E}_Z, \mathbf{E}_X, \mathbf{E}_Y$ de (2.2). Le vecteur d'état est $\boldsymbol{v} = \begin{pmatrix} |v_1| \\ |v_2| e^{i\theta} \end{pmatrix}$. Des calculs triviaux montrent que :

$$|\mathbf{A}_t(1)\boldsymbol{v}|^2 = \begin{pmatrix} |v_1|^2 \\ |v_2|^2 \\ \frac{1}{2}\left(|v_1|^2 + |v_2|^2 + 2|v_1||v_2|\cos(\theta)\right) \\ \frac{1}{2}\left(|v_1|^2 + |v_2|^2 - 2|v_1||v_2|\cos(\theta)\right) \\ \frac{1}{2}\left(|v_1|^2 + |v_2|^2 + 2|v_1||v_2|\sin(\theta)\right) \\ \frac{1}{2}\left(|v_1|^2 + |v_2|^2 - 2|v_1||v_2|\sin(\theta)\right) \end{pmatrix}. \tag{2.12}$$

Par conséquent, $|\mathbf{A}_t(1)\boldsymbol{v}|^2$ donne $|v_1|^2$, $|v_2|^2$, $|v_1||v_2|\cos(\theta)$ et $|v_1||v_2|\sin(\theta)$. À partir de là, deux cas se présentent :

- Si $|v_1| = 0$ ou $|v_2| = 0$ alors, connaître $|v_1|$ et $|v_2|$ suffit car $\begin{pmatrix} |v_1| \\ |v_2| \end{pmatrix}$ est identique à $\boldsymbol{v}$ à une phase globale près. Il n'est donc pas nécessaire de calculer θ.

- Si $|v_1||v_2| > 0$ alors, on peut déduire $\cos(\theta)$ et $\sin(\theta)$ à partir des quantités définies ci-dessus et obtenir θ. Nous connaissons donc tous les paramètres de $\boldsymbol{v}$.

Supposons maintenant que la récupération de l'état est possible pour $n_{qb}-1$ qubits, c'est-à-dire qu'il existe une fonction $f_{n_{qb}-1}$ telle que, pour un vecteur $\boldsymbol{w}$ avec $2^{n_{qb}-1}$ éléments, le vecteur $f_{n_{qb}-1}\left(|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}|^2\right)$ est égal à $\boldsymbol{w}$ à une phase globale près. Soit $\boldsymbol{v}$ un vecteur de taille $d = 2^{n_{qb}}$ (il n'est pas nécessaire qu'il soit de norme unitaire). Nous divisons $\boldsymbol{v}$ en deux vecteurs de $2^{n_{qb}-1}$ éléments : $\boldsymbol{w}_1$ et $\boldsymbol{w}_2$: $\boldsymbol{v} = \begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_2\end{bmatrix}$. Montrons comment $\boldsymbol{v}$ peut être récupéré à une phase globale près à partir de $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$ en utilisant le fait que $\boldsymbol{w}_1$ et $\boldsymbol{w}_2$ peuvent être récupérés à partir de $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_1|^2$ et $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_2|^2$ à des phases globales près en utilisant $f_{n_{qb}-1}$. Nous commençons par comparer $\mathbf{A}_t(n_{qb}-1)$ avec $\mathbf{A}_t(n_{qb})$:

$\mathbf{A}_t(n_{qb}-1) = \begin{bmatrix}\mathbf{E}_{s_1}^*\\ \vdots \\ \mathbf{E}_{s_{2n_{qb}-1}}^*\end{bmatrix}$ où les valeurs $s_1, ..., s_{2n_{qb}-1}$ sont données par (2.11). on remarque aussi que :

$$\mathbf{A}_t(n_{qb}) = \begin{bmatrix}\mathbf{E}_{Zs_1}\\ \vdots \\ \mathbf{E}_{Zs_{2n_{qb}-1}}\\ \mathbf{E}_{X\ldots X}\\ \mathbf{E}_{YX\ldots X}\end{bmatrix} \tag{2.13}$$

où Zs_k est la chaîne de caractères composée de Z suivi de s_k. Avec la définition de $\mathbf{E}$ de la section 2.1.1, on a :

$$\mathbf{E}_{Zs_k} = \mathbf{E}_Z \otimes \mathbf{E}_{s_k} = \begin{bmatrix}\mathbf{E}_{s_k} & \mathbf{0}\\ \mathbf{0} & \mathbf{E}_{s_k}\end{bmatrix} \forall k. \tag{2.14}$$

Soit k un entier entre 1 et $2n_{qb}-1$. Les équations (2.13) et (2.14) donnent :

$$|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2_{i_k} = \left|\begin{bmatrix}\mathbf{E}_{s_k} & \mathbf{0}\\ \mathbf{0} & \mathbf{E}_{s_k}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_2\end{bmatrix}\right|^2 = \begin{bmatrix}|\mathbf{E}_{s_k}\boldsymbol{w}_1|^2\\ |\mathbf{E}_{s_k}\boldsymbol{w}_2|^2\end{bmatrix} \tag{2.15}$$

où $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2_{i_k}$ est le vecteur qui contient les éléments de $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$ avec des indices compris entre $(k-1)d+1$ et kd. Par ailleurs, en utilisant les mêmes notations pour $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_l|^2$ avec l qui vaut 1 ou 2, on a :

$$|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_l|^2_{i_k} = |\mathbf{E}_{s_k}\boldsymbol{w}_l|^2. \tag{2.16}$$

D'après (2.16) et (2.15), il est apparent que tous les éléments de $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_l|^2_{i_k}$ sont dans $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2_{i_k}$ $\forall k \in \{1, ..., 2n_{qb}-1\}$. Puisque $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_l|^2_{i_k}$ $\forall k \in \{1, ..., 2n_{qb}-1\}$ parcourt tout le vecteur $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_l|^2$ nous avons montré que $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_l|^2$ est connu à partir d'une partie des mesures ($|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$) pour $l=1$ et $l=2$.

En utilisant l'hypothèse de récurrence, nous pouvons appliquer $f_{n_{qb}-1}$ aux quantités connues $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_1|^2$ et $|\mathbf{A}_t(n_{qb}-1)\boldsymbol{w}_2|^2$ pour obtenir $\boldsymbol{w}_1$ et $\boldsymbol{w}_2$ à des phases globales près. Appelons nos estimations $\widehat{\boldsymbol{w}_1}$ et $\widehat{\boldsymbol{w}_2}$, $\boldsymbol{w}_1 = e^{i\theta_1}\widehat{\boldsymbol{w}_1}$ et $\boldsymbol{w}_2 = e^{i\theta_2}\widehat{\boldsymbol{w}_2}$. Il nous suffit maintenant de connaître $\theta_2 - \theta_1$ pour connaître $\boldsymbol{v}$ à une phase globale près.

Calculons donc $\theta_2 - \theta_1$ à partir des $2d$ derniers éléments de $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$. Nous définissons $\boldsymbol{L_m}$ comme le vecteur colonne contenant les $2d$ derniers éléments

$$\boldsymbol{L_m} = \left| \begin{bmatrix} \mathbf{E}^*_{XX...X} \\ \mathbf{E}^*_{YX...X} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} \right|^2 = \left| \begin{bmatrix} \mathbf{E}^*_{X} \otimes \mathbf{E}^*_{X...X} \\ \mathbf{E}^*_{Y} \otimes \mathbf{E}^*_{X...X} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} \right|^2$$

où, à gauche, les chaînes de caractère $XX...X$, $YX...X$ ont n_{qb} caractères, et, à droite, $X...X$ a $n_{qb}-1$ caractères. En remplaçant $\mathbf{E}_X$ et $\mathbf{E}_Y$ par les valeurs qu'ils ont dans la section 2.1.1 et en calculant les produits tensoriels, on a :

$$\boldsymbol{L_m} = \left| \tfrac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{E}^*_{X...X}\boldsymbol{w}_1 + \mathbf{E}^*_{X...X}\boldsymbol{w}_2 \\ \mathbf{E}^*_{X...X}\boldsymbol{w}_1 - \mathbf{E}^*_{X...X}\boldsymbol{w}_2 \\ \mathbf{E}^*_{X...X}\boldsymbol{w}_1 - i\mathbf{E}^*_{X...X}\boldsymbol{w}_2 \\ \mathbf{E}^*_{X...X}\boldsymbol{w}_1 + i\mathbf{E}^*_{X...X}\boldsymbol{w}_2 \end{bmatrix} \right|^2 = \tfrac{1}{2} \left| \begin{bmatrix} \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_1}e^{i\theta_1} + \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_2}e^{i\theta_2} \\ \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_1}e^{i\theta_1} - \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_2}e^{i\theta_2} \\ \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_1}e^{i\theta_1} - i\mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_2}e^{i\theta_2} \\ \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_1}e^{i\theta_1} + i\mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_2}e^{i\theta_2} \end{bmatrix} \right|^2.$$

Nous introduisons les notations suivantes :

$$\begin{aligned} \boldsymbol{m} &= \frac{1}{2}|\mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_1}|^2 + \frac{1}{2}|\mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_2}|^2 \\ \boldsymbol{d_c} &= \overline{\mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_1}} \odot \mathbf{E}^*_{X...X}\widehat{\boldsymbol{w}_2} \\ \boldsymbol{d}(\theta) &= \cos(\theta)Re(\boldsymbol{d_c}) - \sin(\theta)Im(\boldsymbol{d_c}) \end{aligned} \tag{2.17}$$

où $\bar{\cdot}$ est le conjugué et on rappelle que $\odot$ représente le produit élément par élément. $\widehat{\boldsymbol{w}_1}$ et $\widehat{\boldsymbol{w}_2}$ sont connus (à partir de $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$) ; donc $\boldsymbol{m}$ et $\boldsymbol{d_c}$ sont connus et $\boldsymbol{d}(\theta)$ peut être calculé pour tout $\theta \in [0, 2\pi]$. Ré-écrivons $\boldsymbol{L_m}$ comme une fonction de $(\theta_2 - \theta_1)$ en utilisant ces quantités :

$$\boldsymbol{L_m}(\theta_2 - \theta_1) = \begin{bmatrix} \boldsymbol{m} + \boldsymbol{d}(\theta_2 - \theta_1) \\ \boldsymbol{m} - \boldsymbol{d}(\theta_2 - \theta_1) \\ \boldsymbol{m} + \boldsymbol{d}(\theta_2 - \theta_1 - \pi/2) \\ \boldsymbol{m} - \boldsymbol{d}(\theta_2 - \theta_1 - \pi/2) \end{bmatrix}. \tag{2.18}$$

Nous cherchons à calculer $\theta_2 - \theta_1$ à partir de $\boldsymbol{L_m}$ (qui est connu grâce aux mesures). Remarquons d'abord que, d'après la définition de $\boldsymbol{d}(\theta)$ dans (2.17) si $\boldsymbol{d_c}$ est nul sur chaque composante, alors $\boldsymbol{d}(\theta_2 - \theta_1)$ est également nul sur chaque composante (ce qui signifie qu'il ne dépend pas de $\theta_2 - \theta_1$) et que $\boldsymbol{L_m}$ est simplement $\boldsymbol{m}$ répété 4 fois (voir (2.18)). Par conséquent, estimer $\theta_2 - \theta_1$ (et $\boldsymbol{v}$) à partir de $\boldsymbol{L_m}$ est impossible. Cependant, nous montrons ci-après qu'il s'agit du seul cas où $\theta_2 - \theta_1$ ne peut être récupéré à partir de $\boldsymbol{L_m}$. Et l'ensemble des $\boldsymbol{v}$ qui font que cela se produit a une mesure nulle.

Supposons qu'au moins un élément de $\boldsymbol{d_c}$ n'est pas nul. Appelons k son indice et d_k l'élément non nul correspondant (nous prenons l'élément qui a le module le plus élevé), et appelons $d_k(\theta_2 - \theta_1)$ et m_k les k-ième éléments de $\boldsymbol{d}(\theta_2 - \theta_1)$ et $\boldsymbol{m}$ respectivement. Nous avons seulement besoin des k-ème et $(k+d)$-ème éléments de $\boldsymbol{L_m}$ dont les expressions sont $m_k + \cos(\theta_2 - \theta_1)Re(d_k) - \sin(\theta_2 - \theta_1)Im(d_k)$ et $m_k + \sin(\theta_2 - \theta_1)Re(d_k) + \cos(\theta_2 - \theta_1)Im(d_k)$. Ces éléments connus peuvent être placés dans un vecteur colonne et réécrits comme suit

$$\begin{pmatrix} Re(d_k) & -Im(d_k) \\ Im(d_k) & Re(d_k) \end{pmatrix} \begin{pmatrix} \cos(\theta_2 - \theta_1) \\ \sin(\theta_2 - \theta_1) \end{pmatrix}. \tag{2.19}$$

La matrice 2×2 de gauche est connue (puisque d_k est connue) et inversible (puisque son déterminant est $|d_k|^2 > 0$). Par conséquent, $\theta_2 - \theta_1$ peut être calculé (car nous avons son sinus et son cosinus) à partir de deux éléments de $\boldsymbol{L_m}$ (donc deux probabilités).

On pourrait s'arrêter là et obtenir une estimation θ_d de $\theta_2 - \theta_1$ qui est calculée à partir de deux éléments de $\boldsymbol{L_m}$. Mais, en pratique, les probabilités de l'échantillon donnent une estimation imparfaite de $\boldsymbol{L_m}$ que nous appelons $\widehat{\boldsymbol{L_m}}$. Afin d'être robuste aux erreurs, nous proposons donc de trouver l'angle $\widehat{\theta_2 - \theta_1}$ qui minimise $||\boldsymbol{L_m}(\theta_2 - \theta_1) - \widehat{\boldsymbol{L_m}}||$. De cette manière, nous utilisons toutes les probabilités empiriques et pas seulement deux. Nous utilisons un algorithme BFGS quasi-Newton [Bro70] (mis en œuvre avec la fonction fminunc dans le logiciel numérique Matlab) initialisé à θ_d. L'optimisation s'arrête lorsque le pas est inférieur à 10^{-30}. Techniquement, avec

cette optimisation, l'algorithme n'est plus explicite, mais, comme elle ne fait intervenir qu'un seul paramètre, elle est très rapide et améliore les performances de manière significative, c'est pourquoi nous avons choisi de l'effectuer quand même. Pour avoir un algorithme de forme fermée, on pourrait utiliser θ_d au lieu de calculer $\widehat{\theta_2 - \theta_1}$ ou alors utiliser un algorithme d'optimisation avec un nombre fixe d'étapes pour calculer $\widehat{\theta_2 - \theta_1}$.

Prenons maintenant un peu de recul et résumons ce que nous avons prouvé dans cette section :

- La récupération de l'état (à une phase globale près) à partir des mesures est possible pour $n_{qb} = 1$.
- En supposant qu'elle est possible pour $n_{qb} - 1$ nous avons montré qu'elle est également possible pour n_{qb} sauf si l'état se trouve dans un ensemble de mesure nulle.

En utilisant les deux résultats précédents, nous pouvons construire un algorithme récursif qui récupère $\boldsymbol{v}$ à partir des mesures. Il fonctionne pour tous les $\boldsymbol{v}$ sauf sur l'union d'un nombre fini d'ensembles de mesure nulle. Cette union est donc également de mesure nulle. L'estimation donnée par cet algorithme récursif est appelée $\widehat{\boldsymbol{v}}_{rec}$.

2.3.3 Discussion sur le nombre de probabilités utilisées

L'algorithme récursif de la section précédente s'appelle deux fois pour chaque passage de n_{qb} à $n_{qb} - 1$. Cela signifie que pour un n_{qb} donné, il est appelé une fois avec n_{qb} qubits, deux fois avec $n_{qb} - 1$ qubits, ..., $2^{n_{qb}-1}$ fois avec un qubit.

Pour un qubit, l'état est récupéré à l'aide de (2.12). Cette équation implique six probabilités, dont seulement quatre sont nécessaires (nous pourrions obtenir le même résultat sans utiliser les quatrième et sixième éléments de $|\mathbf{A_t}(1)\boldsymbol{v}|^2$).

Pour $n_{qb} > 1$, avant d'appeler la fonction récursive avec un qubit de moins, nous calculons $\theta_2 - \theta_1$ en utilisant (2.18). Cela implique $2 \times 2^{n_{qb}}$ probabilités dont seulement 2 sont strictement nécessaires pour la première estimation θ_d.

Le nombre minimum de probabilités nécessaires est $4 \times 2^{n_{qb}-1} + 2\sum_{q=2}^{n_{qb}} 2^{n_{qb}-q} = 2d + 2(2^{n_{qb}-1} - 1) = 3d - 2$. De plus, si l'on tient compte du fait que $\boldsymbol{v}$ a une norme unitaire, alors, une des probabilités selon l'axe Z (qui sont toutes utilisées) devient redondante, et ce nombre devient $3(d-1)$.

En pratique, toutes les probabilités sont utilisées afin de minimiser l'impact des erreurs statistiques sur les probabilités. Mais, si nous voulions supprimer des lignes de $\mathbf{A}_t$ dans (2.11) et ne conserver que $3(d-1)$ d'entre elles, nous pourrions encore effectuer la QST. Cela serait une mauvaise idée cependant car $\mathbf{A}_t$ ne serait plus la concaténation de matrices unitaires, ce qui veut dire que nos mesures ne pourraient plus être effectuées en parallèle. Et, en pratique, l'estimation finale de l'état serait moins robuste aux erreurs sur les probabilités empiriques ; et l'implémentation physique ne serait pas plus facile à mettre en place, car l'estimation des $3(d-1)$ probabilités à conserver nécessite de toute façon la réalisation de toutes les $2n_{qb} + 1$ mesures à effectuer.

2.3.4 Comparaison avec la littérature

Résumons les principales caractéristiques de notre deuxième algorithme QST :

- Il utilise $(2n_{qb} + 1)d$ probabilités qui peuvent être obtenues en faisant la moyenne des résultats de $2n_{qb} + 1$ mesures parallèles non intriquées.
- Les mesures sont injectives en dehors d'un ensemble connu de mesure nulle.

- L'algorithme qui reconstruit l'état est explicite.

Ces caractéristiques sont très similaires à celles de Goyeneche et al. [GCE+15]. L'avantage de notre méthode est que les mesures qu'elle utilise sont non intriquées. Son inconvénient est qu'elle nécessite $2n_{qb}+1$ types de mesures ce qui est supérieur à quatre (sauf pour le cas trivial $n_{qb}=1$). C'est le prix à payer pour n'utiliser que des mesures non intriquées. Nous n'avons pas trouvé d'algorithme simple et explicite qui fonctionne avec moins de types de mesures non intriquées.

L'approche parcimonieuse plus générale de [GLF+10] nécessite $O(rd\log(d)^2)$ probabilités pour estimer l'état où r est le rang de la matrice densité, qui vaut 1 dans le cas d'un état pur. Ces probabilités pourraient être obtenues en faisant la moyenne des résultats de $O(\log(d)^2)$ différentes mesures non intriquées. Nous faisons mieux ici puisque nous n'utilisons que des moyennes de $2n_{qb}+1 = O(\log(d))$ types de mesures. Nous avons également l'avantage de fournir un algorithme explicite contrairement à la méthode de [GLF+10] qui est très générale (elle fonctionne pour les états mélange et tout type de mesure), mais utilise un algorithme d'optimisation et ne fournit pas de preuve d'injectivité.

2.4 Tomographie d'état par maximisation de la vraisemblance

2.4.1 Idée principale

Les algorithmes des sections 2.2 et 2.3 nous donnent des estimations de l'état $\boldsymbol{v}$, appelées $\widehat{\boldsymbol{v}}_{pc}$ et $\widehat{\boldsymbol{v}}_{rec}$ respectivement. $\widehat{\boldsymbol{v}}_{pc}$ est la solution du problème de la QST avec une contrainte $(\text{rang}(\mathbf{U})=1)$ relâchée, elle peut être inexacte même en l'absence d'erreurs dans les probabilités empiriques. L'algorithme de la section 2.3.2 qui calcule $\widehat{\boldsymbol{v}}_{rec}$ est également imparfait. Il s'appuie fortement sur les mesures selon $Z...Z$, $Z..ZX$ et $Z..ZY$ (utilisées $2^{n_{qb}-1}$ fois pour un qubit à la fin de l'arbre récursif pour calculer tous les modules et la moitié des différences de phases) et n'utilise presque pas les mesures selon $X...X$ et $YX..X$ (utilisées une seule fois pour calculer une différence de phase $(\theta_2-\theta_1)$ avec (2.18)). Or tous les types de mesures contiennent autant d'information sur $\boldsymbol{v}$ (les probabilités théoriques sont dans $|\mathbf{E}\boldsymbol{v}|^2$, et seul $\mathbf{E}$ change), un algorithme optimal ne devrait pas les utiliser d'une telle façon.

Pour ces raisons, les méthodes d'estimation des sections 2.2 et 2.3 sont complétées ci-après par une dernière optimisation afin de les rendre plus précises. À cette fin, nous adoptons une approche fondée sur le maximum de vraisemblance ("maximum likelihood" ou ML) :

$$(\widehat{\boldsymbol{x}},\widehat{\boldsymbol{y}}) = \underset{\boldsymbol{x},\boldsymbol{y}\ \text{t.q.}\ ||\boldsymbol{x}||_2+||\boldsymbol{y}||_2<1}{\arg\min} \mathcal{L}_{(\boldsymbol{x},\boldsymbol{y})}(\widehat{\boldsymbol{p}}) \tag{2.20}$$

où $\widehat{\boldsymbol{p}}$ est le vecteur qui contient les probabilités empiriques mesurées et $\mathcal{L}_{(\boldsymbol{x},\boldsymbol{y})}(\widehat{\boldsymbol{p}})$ doit être compris comme l'opposé de la log-vraisemblance (ou log-vraisemblance négative) des probabilités empiriques $\widehat{\boldsymbol{p}}$ si l'état réel est $\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y})$, avec $\boldsymbol{x}$ et $\boldsymbol{y}$ définis ci-après. Dans l'ensemble du document, chaque fois que nous écrivons "log-vraisemblance négative" (ou $\mathcal{L}$), nous voulons dire "l'opposé de la log-vraisemblance à une constante additive près". Cette constante n'a pas d'importance puisque la log-vraisemblance négative sera minimisée. Le vecteur $\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y})$ par rapport auquel $\mathcal{L}$ sera minimisée est défini comme suit :

$$\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y}) = \begin{pmatrix} \sqrt{1-||\boldsymbol{x}||_2^2-||\boldsymbol{y}||_2^2} \\ x_1+iy_1 \\ \vdots \\ x_{d-1}+iy_{d-1} \end{pmatrix}.$$

$\boldsymbol{x}$ et $\boldsymbol{y}$ sont des vecteurs de taille $d-1$ qui représentent les parties réelles et imaginaires des derniers éléments de $\boldsymbol{v}$. La contrainte dans (2.20) est $r^2<1$ (avec $r=\sqrt{||\boldsymbol{x}||_2^2+||\boldsymbol{y}||_2^2}$) et non

$r^2 \leq 1$ car l'optimisation est plus facile sur un ensemble ouvert. Nous atténuons l'effet de cette contrainte inexacte en permutant la première composante de $\boldsymbol{v}$ et la composante de $\boldsymbol{v}$ ayant le module le plus élevé au point initial de l'optimisation. Ainsi, nous nous assurons que r^2 ne sera pas proche de 1, sauf si le point initial est totalement aberrant. Les probabilités empiriques et les colonnes de $\mathbf{A}$ sont permutées de la même manière. Ces changements sont limités à l'algorithme d'optimisation.

Puisque l'ensemble d'optimisation est ouvert, nous pouvons changer les variables afin de supprimer complètement la contrainte :
$\boldsymbol{x}' = \frac{tan(\frac{\pi}{2}r)}{r}\boldsymbol{x}, \boldsymbol{x} = \frac{\frac{2}{\pi}atan(r')}{r'}\boldsymbol{x}'$ donc $\boldsymbol{y}' = \frac{tan(\frac{\pi}{2}r)}{r}\boldsymbol{y}, \boldsymbol{y} = \frac{\frac{2}{\pi}atan(r')}{r'}\boldsymbol{y}'$ (avec $r' = \sqrt{||\boldsymbol{x}'||_2^2 + ||\boldsymbol{y}'||_2^2}$).
Le nouveau problème d'optimisation sur $\boldsymbol{x}'$ et $\boldsymbol{y}'$ n'a pas de contrainte, car lorsque r' couvre tout l'espace, r reste strictement inférieur à un. L'équation (2.20) est donc remplacée par :

$$(\widehat{\boldsymbol{x}'}, \widehat{\boldsymbol{y}'}) = \underset{\boldsymbol{x}',\boldsymbol{y}'}{\arg\min} \mathcal{L}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}). \tag{2.21}$$

Pour résoudre (2.21), nous utilisons à nouveau l'algorithme BFGS [Bro70] où les expressions analytiques des gradients sont fournies. L'algorithme s'arrête lorsque la norme du pas d'optimisation est inférieure à 10^{-30}. Comme dans la plupart des méthodes d'optimisation non convexe, nous avons besoin d'un bon point d'initialisation, nous utilisons soit $\widehat{\boldsymbol{v}}_{pc}$ soit $\widehat{\boldsymbol{v}}_{rec}$. Le point le plus vraisemblable pour $\boldsymbol{v}$ est $\widehat{\boldsymbol{v}}_{ml} = \boldsymbol{v}(\widehat{\boldsymbol{x}'}, \widehat{\boldsymbol{y}'})$, avec les $\widehat{\boldsymbol{x}'}, \widehat{\boldsymbol{y}'}$ défini dans (2.21).

Il ne reste plus qu'à définir l'expression de la log-vraisemblance négative $\mathcal{L}$ par rapport à $\boldsymbol{v}$. Dans les deux sous-sections suivantes, nous donnerons deux expressions pour la log-vraisemblance normalisée :$\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$ et $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$.

2.4.2 Vraisemblance exacte

La formule de la vraisemblance d'une mesure quantique est donnée dans [HŘFJ04] (pour un état mixte représenté par un opérateur densité ρ que nous devons remplacer ici par $\boldsymbol{v}\boldsymbol{v}^*$). Elle se résume à :

$$\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = -\sum_{k=1}^{n_{prob}} n_k log((|\mathbf{A}\boldsymbol{v}(\boldsymbol{x}',\boldsymbol{y}')|^2)_k). \tag{2.22}$$

$(|\mathbf{A}\boldsymbol{v}(\boldsymbol{x}',\boldsymbol{y}')|^2)_k$ est le k-ème élément de $|\mathbf{A}\boldsymbol{v}(\boldsymbol{x}',\boldsymbol{y}')|^2$, $\mathbf{A}$ est soit $\mathbf{A}_s$ soit $\mathbf{A}_t$, n_k est le nombre de réalisations du k-ème résultat possible, c.à.d du k-ème élément de $\widehat{\boldsymbol{p}}$ (soit $\widehat{\boldsymbol{p_s}}$, soit $\widehat{\boldsymbol{p_t}}$) multiplié par le nombre (n_c) de répétitions de chaque type de mesure sur l'état mesuré, et n_{prob} est le nombre de probabilités mesurées, c'est aussi le nombre de lignes de $\mathbf{A}$.

Pour parvenir à ce résultat, il faut considérer les nombres de mesures comme les réalisations d'une variable aléatoire multinomiale. Il ne s'agit pas d'une approximation, c'est pourquoi nous appelons cette vraisemblance "exacte".

2.4.3 Régularisation gaussienne

Dans cette sous-section, nous utilisons le théorème de la limite centrale (TLC) pour approximer les probabilités empiriques comme la réalisation d'une distribution normale multivariée. Cette méthode est appropriée, car le vecteur $\widehat{\boldsymbol{p}}$ dont nous voulons calculer la vraisemblance est la moyenne des réalisations indépendantes de la même variable aléatoire (c'est le cas d'application du TLC). L'espérance de $\widehat{\boldsymbol{p}}$ est le vecteur de probabilités théoriques $\boldsymbol{p}(\boldsymbol{x}',\boldsymbol{y}')$ qui dépend de l'état. Définissons $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}') = \widehat{\boldsymbol{p}} - \boldsymbol{p}(\boldsymbol{x}',\boldsymbol{y}')$ et $\underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ est $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ avec le dernier élément enlevé (aucune information n'est perdue puisque la somme des éléments de $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ est égale à 0). Dans l'Annexe A.2.1, nous montrons que si n_c est le nombre de fois que les mesures ont été moyennées, alors $\sqrt{n_c}\underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ suit asymptotiquement $(n_c \to +\infty)$ une distribution normale

multivariée à moyenne nulle. Sa matrice de covariance $\boldsymbol{\Sigma}$ est calculée dans l'Annexe A.2.1. $\boldsymbol{\Sigma}$ dépend des probabilités théoriques, nous voulons supprimer cette dépendance. Nous remplaçons donc $\boldsymbol{\Sigma}$ par $\widetilde{\boldsymbol{\Sigma}}$ qui est une approximation de la matrice de covariance qui utilise $\widetilde{\boldsymbol{p}} = \frac{\widehat{\boldsymbol{p}}+\frac{5}{n_c}}{1+\frac{5d}{n_c}}$ comme une approximation de $\boldsymbol{p}$, cette régularisation a deux avantages (i) elle rend le critère plus "lisse" et facile (moins d'itérations et meilleure probabilité de trouver le minimum global) à minimiser (ii) elle modélise bien des erreurs de décohérence (qui font que les états mesurés sont des états mélanges, voir section 2.5.5) ou de "bit-flip readout" à la mesure (qui font qu'on a une probabilité p_{flip} de mesurer le mauvais bit sur chaque bit, cette erreur rend possible de mesurer des résultats qui sont censés avoir une probabilité nulle de se produire) qui ne sont pas prises en compte dans le modèle de l'état et des mesures.

Avec ce modèle gaussien régularisé, nous obtenons l'approximation suivante pour la log-vraisemblance négative :

$$\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = n_c \underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')^T \widetilde{\boldsymbol{\Sigma}}^{-1} \underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}'). \tag{2.23}$$

L'Annexe A.2.1 montre également que cette équation peut se réduire à :

$$\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = n_c \sum_{k=1}^{n_{prob}} \frac{\varepsilon_k(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')^2}{\widetilde{p}_k}. \tag{2.24}$$

où n_{prob} est le nombre de probabilités mesurées (qui correspond au nombre de ligne de $\mathbf{A}$) et $\varepsilon_k(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ est le k-ème élément de $\varepsilon(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$.

Cette log-vraisemblance est le résultat de deux approximations qui ne sont vraies que lorsque $n_c \to +\infty$: (i) nous avons approximé $\underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ comme la réalisation d'un vecteur aléatoire gaussien et (ii) nous avons utilisé une approximation pour $\boldsymbol{\Sigma}$. En pratique, l'approximation résultante est plus lisse et plus facile à minimiser que $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$ si le point d'initialisation n'est pas suffisamment bon (comme nous le montrerons dans la section 2.5.3). Cependant, avec une bonne initialisation, l'état qui minimise $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$ devrait (si l'erreur est effectivement multinomiale) être plus proche de l'état réel que celui qui minimise $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$. Plus n_c est petit, plus la différence est marquée. Ce sera montré dans la section 2.5.2.

2.4.4 Algorithme mixte

Comme indiqué ci-dessus, $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ est censé être plus facile à minimiser, mais le minimum de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ est censé être une meilleure estimation. Une bonne façon de combiner les deux avantages est de commencer le processus d'optimisation en minimisant $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ et de le terminer en minimisant $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$. En pratique, nous exécutons à nouveau l'algorithme BFGS [Bro70] sur $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ pendant 100 itérations à partir du point d'initialisation des sections 2.2 ou 2.3, ce qui donne $\widehat{\boldsymbol{v}}_{inter}$. Ensuite, nous exécutons l'algorithme BFGS sur $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ en commençant par $\widehat{\boldsymbol{v}}_{inter}$ et ne nous arrêtons que lorsqu'un minimum local (que l'on espère être global) a été trouvé.

2.5 Test des algorithmes de QST en simulations

Nous avons choisi de tester nos algorithmes de QST indépendamment dans un premier temps pour vérifier qu'ils sont intéressants en tant que tels. L'autre objectif des tests de la présente section est de faire des choix (informés par des simulations) qui n'ont pas été faits dans les parties précédentes. Ces choix sont : (i) la configuration favorite entre celle de la section 2.2 et celle de la section 2.3 ($n_t = 4$ et $n_t = 2n_{qb}+1$) (ii) le nombre d'itération de PhaseCut (iii) lequel des trois types d'algorithmes de vraisemblance on privilégie.

Tous nos tests sont d'abord réalisés sur $n_{qb} = 7$ qubits, et comme nous l'avons expliqué dans la section 1.1.7 c'est déjà beaucoup ; il existe des systèmes avec plus de qubits, mais ils sont incapables de réaliser des états purs qui intriquent plus de 5-6 qubits. Nous ne testons pas nos algorithmes avec moins de 7 qubits dans un premier temps car nous voulons nous placer dans la pire des configuration pour choisir le nombre d'itérations de phaseCut et étudier les écarts entre les algorithmes de maximum de vraisemblance.

Afin de comparer de façon équitable les configurations avec $n_t = 4$ et $n_t = 2n_{qb} + 1$ types de mesures, nous gardons le nombre total de mesures $n_c n_t$ soit le même sur les deux configurations.

2.5.1 Performances des deux algorithmes d'initialisation

Les sections 2.2 et 2.3 décrivent en détail deux méthodes de QST qui sont utilisées pour l'initialisation des algorithmes ML. La présente section vise à estimer la précision de ces méthodes et à les comparer dans la mesure du possible. L'algorithme récursif de la section 2.3 ne fonctionne que pour un ensemble spécifique de types de mesures, mais il est explicite et ne nécessite pas un nombre indéfini d'itérations pour converger, contrairement à PhaseCut défini dans la section 2.2. Nous n'avons présenté PhaseCut que pour la configuration avec les quatre types de mesures différents décrits dans la section 2.2.1, mais elle peut être appliquée à tous les types de mesures. En particulier, nous pourrions l'appliquer à la configuration avec $2n_{qb} + 1$ types de mesures de la section 2.3.1.

Dans la présente section, nous testons PhaseCut et l'algorithme récursif sur 50 états purs à 7 qubits générés aléatoirement. Chaque partie réelle et imaginaire de chaque composante du vecteur d'état est fixée à un nombre réel pseudo-aléatoire suivant une loi gaussienne, puis, le vecteur est normalisé. Les deux ensembles de types de mesures des sections 2.2.1 et 2.3.1 sont pris en compte. Ils contiennent respectivement 4 et $2 \times 7 + 1 = 15$ types de mesures. Nous testons ces algorithmes avec deux nombres de mesures total (produit de n_c, nombre de copie de l'état préparées pour chaque type de mesure et de n_t, le nombre de type de mesures) $n_c n_t$: 5000 et 500000. Ainsi, chacun des 4 types de mesures de la configuration de la section 2.2.1 est effectué $n_c = 1250$ ou $n_c = 125000$ fois et chacun des 15 types de mesures de la configuration de la section 2.2.1 est effectué $n_c = 333$ ou $n_c = 33333$ fois. Les deux algorithmes sont testés sur un ensemble de 50 états pur aléatoires.

La métrique utilisée pour quantifier la proximité entre $\widehat{\boldsymbol{v}}$ et le vecteur réel $\boldsymbol{v}$ à un facteur de phase près est

$$\mu_s = \frac{1}{\sqrt{2}} ||\boldsymbol{v} - \widehat{\boldsymbol{v}}.e^{-i\xi}||_2 \tag{2.25}$$

avec ξ l'angle qui minimise notre métrique : $e^{i\xi} = \frac{\boldsymbol{v}^*\widehat{\boldsymbol{v}}}{|\boldsymbol{v}^*\widehat{\boldsymbol{v}}|}$. Nous appelons μ_s cette erreur dans le reste du document. μ_s est maximal pour les états orthogonaux (il vaut alors 1), et minimal pour les états qui sont les mêmes à une phase globale près (il vaut alors 0). Une mesure plus largement utilisée dans la littérature est la fidélité (voir la section 9.2.2 dans [NC00]) $f = |\boldsymbol{v}^*\widehat{\boldsymbol{v}}|$. On peut montrer que $f = (1 - \mu_s^2)$, voir Annexe A.4. Nous n'utilisons pas la fidélité, car les petites erreurs sont trop proches de 1 et difficilement discernables. Nous pensons également que notre métrique est plus intuitive : $\mu_s = 0{,}10$ peut être compris comme une erreur de 10% alors que la signification de $f = 0{,}99$ n'est pas aussi claire.

La figure 2.4 représente l'erreur de $\widehat{\boldsymbol{v}}_{pc}$ obtenue en utilisant PhaseCut avec 100 à 100000 itérations pour les deux configurations (4 et 15 types de mesures). Avec 15 types de mesures, l'algorithme récursif peut être mis en œuvre (ce n'est pas le cas avec 4 mesures). Nous affichons les plus grandes et les plus petites erreurs de $\widehat{\boldsymbol{v}}_{rec}$ obtenues avec l'algorithme récursif avec des lignes horizontales rouges pour les plus grosses erreurs (en haut) et vertes pour les plus petites (en bas). L'algorithme récursif est (par sa nature) exécuté avec un nombre fixe d'étapes, c'est

pourquoi nous représentons ses erreurs sous forme de lignes horizontales et non de courbes en fonction du nombre d'itérations.

Le but de cette simulation est de voir combien d'itérations de PhaseCut sont nécessaires pour obtenir une bonne estimation de l'état et de comparer les performances de l'algorithme récursif avec celles de PhaseCut, plus polyvalent.

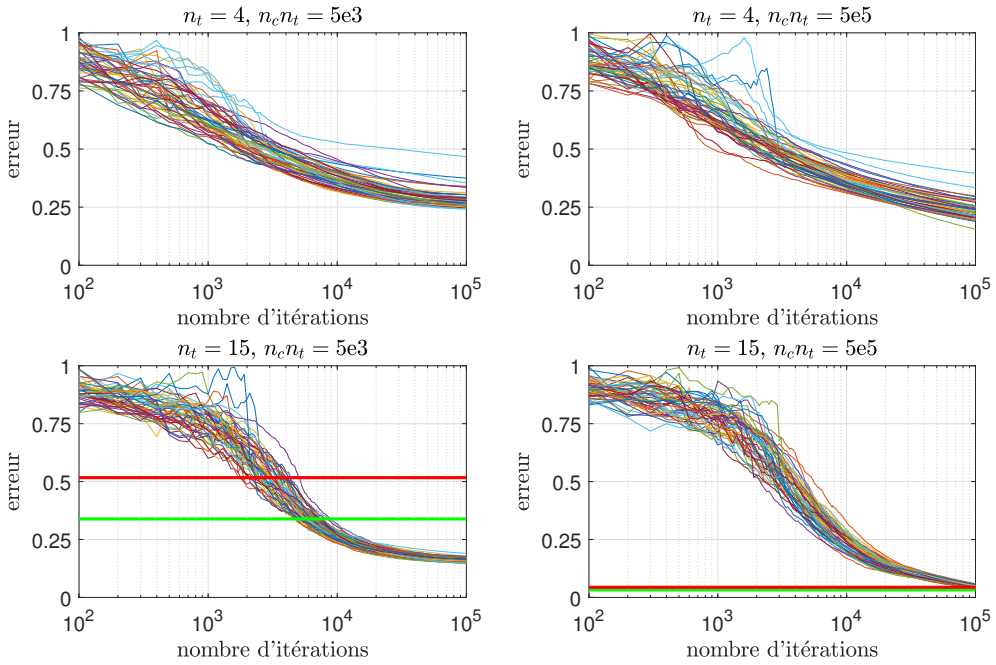


FIGURE 2.4 : Erreurs de QST (définies par (2.25)) des algorithmes d'initialisation, avec $n_t = 4$ et $n_t = 15$ type de mesure, et un nombre total de mesure $n_c n_t$ de 5000 et 500 000. Les lignes horizontales en gras rouges (en haut) et vertes (en bas) représentent les pires et les meilleures erreurs pour l'algorithme récursif (disponible uniquement avec 15 types de mesures) sur les 50 états purs générés aléatoirement. Les autres courbes représentent l'évolution de l'erreur sur les estimations des 50 états avec PhaseCut en fonction du nombre d'itérations.

Avec suffisamment d'itérations ($\sim 10^4$ pour $n_c n_t = 5000$ et $\sim 10^5$ pour $n_c n_t = 500000$) PhaseCut est plus précis que l'algorithme récursif dans les configurations pour lesquelles ils peuvent tous deux être mis en œuvre, mais, comme nous allons le montrer, il est beaucoup plus lent. Chaque itération de PhaseCut est coûteuse, car nous travaillons sur une matrice de carrée qui a autant de lignes que $\mathbf{A}$, c.à.d $dn_t = 512$ pour $n_t = 4$ ou $dn_t = 1920$ pour $n_t = 2n_{qb}+1 = 15$ ($n_{qb} = 7$ donc $d = 128$ dans les deux cas). Avec le logiciel Matlab, sur un processeur à 4 cœurs de 2,11 GHz avec une RAM de 32 Go, chaque itération de PhaseCut prend environ 4 ms pour la configuration avec $n_t = 4$ types de mesures et environ 45 ms pour la configuration avec $n_t = 15$ types de mesures. Dans cette même configuration à 15 types de mesures, l'algorithme récursif prend 200 ms au total. C'est beaucoup plus rapide que PhaseCut, qui s'exécute en quelques minutes, car il nécessite des milliers d'itérations.

2.5.2 Précisions des algorithmes de maximisation de la vraisemblance

Dans la section 2.4, nous avons défini deux estimateurs de vraisemblance, basés sur la maximisation de la vraisemblance. Le premier minimise la véritable log-vraisemblance négative $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ et l'autre minimise une version de la log-vraisemblance négative censée être plus lisse, à savoir $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$. Nous savons que $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ est une approximation de la vraisemblance qui n'est précise

que si le nombre de mesures par type de mesure est suffisamment élevé. Par conséquent, nous nous attendons à ce que le minimum global de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ soit un moins bon estimateur que le minimum global de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ pour un nombre limité de mesures. Afin de vérifier si cela est vrai et de quantifier la différence, nous calculons les erreurs des deux estimateurs lorsqu'ils sont initialisés au vrai $\boldsymbol{v}$, on considère que la valeur de l'optimisation qui en résulte est le maximum global de vraisemblance (le maximum de vraisemblance devrait être proche du vrai $\boldsymbol{v}$ tant que l'erreur n'est pas trop grande). Ce faisant, nous ignorons l'erreur sur le point d'initialisation (à laquelle l'estimation gaussienne régularisée est censée être plus robuste que l'estimation par minimisation de la vraisemblance exacte). Nous calculons également l'erreur pour l'algorithme mixte qui commence par minimiser $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ puis minimise $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$. Ces trois types d'erreurs sont calculés avec 1000 états initiaux générés aléatoirement pour les quatre configurations décrites dans la section 2.5.1 avec $n_t = 4$ ou $n_t = 15$ types de mesures et $n_c n_t = 5000$ ou $n_c n_t = 500000$. Pour chacune des quatre configurations, la fonction de répartition empirique (fdr empirique) est déduite des 1 000 erreurs associées aux états initiaux ; ces fdr sont illustrées à la figure 2.5.

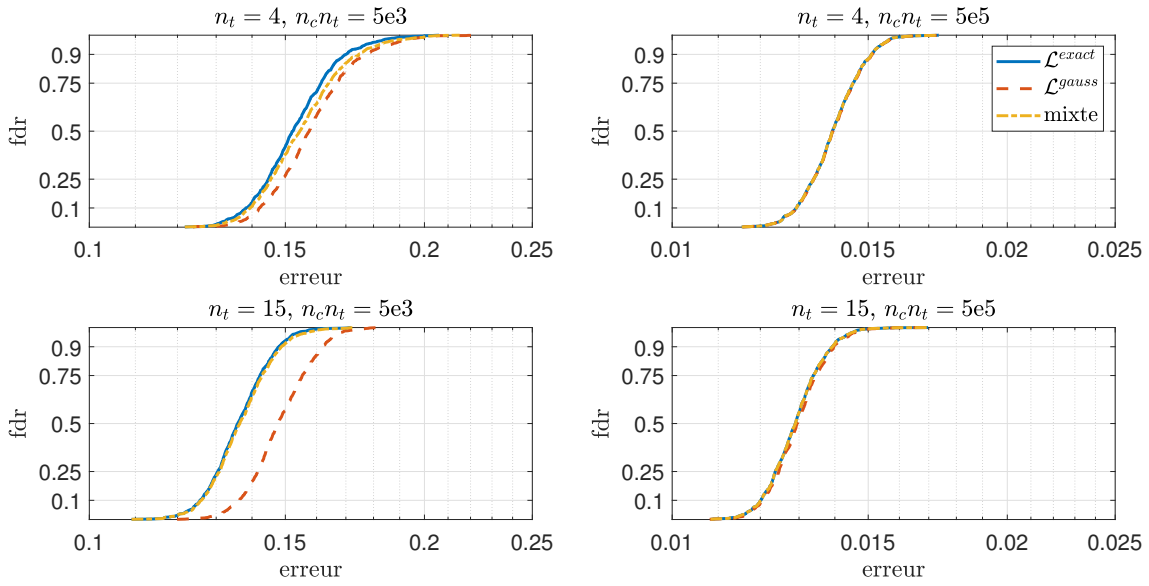


FIGURE 2.5 : fdr des erreurs des trois algorithmes de maximisation de la vraisemblance initialisés sans erreur. Les quatre configurations des quatre sous-plots sont les mêmes que celles de la figure 2.4.

Comme prévu, l'erreur est plus importante avec l'estimation gaussienne de la vraisemblance, et la différence diminue lorsque le nombre de mesures par type de mesure augmente.

Les performances de l'algorithme de minimisation mixte est très proche de celle de l'estimateur qui minimise $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$. Il peut cependant y avoir de petites différences. Il s'avère qu'ils convergent parfois vers des minima proches, mais différents. Cela est dû au fait que la petite erreur commise par les 100 premières itérations de l'algorithme mixte (au cours desquelles $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ est minimisé) peut être suffisante pour affecter le résultat final.

Les différences entre les trois estimateurs ne sont remarquables que pour $n_c n_t = 5000$ avec 15 et 4 mesures différentes (donc $n_c = 333$ ou $n_c = 1250$ mesures par type de mesure).

2.5.3 Robustesse des algorithmes de maximisation de la vraisemblance

Dans la présente section, nous cherchons quelles précisions sur l'état initial sont nécessaires pour s'assurer que les différents algorithmes d'optimisation de la vraisemblance convergent vers une solution raisonnable. C'est ce que nous appelons la "robustesse". Nous comparons les taux

de divergence (notés δ et définis ci-dessous) des algorithmes qui minimisent $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ et $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ ainsi que de l'algorithme mixte. 1000 états $\boldsymbol{v}$ à estimer sont générés aléatoirement (avec la même méthode que celle utilisée pour générer les 50 états de la section 2.5.1). On définit aussi 1000 états initiaux (un par état à estimer) avec une erreur d'initialisation μ_s (erreur entre un état à estimer et l'état initial associé). μ_s varie linéairement de 0 à 1 (le premier état initial est le même que l'état à estimer associé ($\mu_s = 0$), le dernier est orthogonal ($\mu_s = 1$)). On appelle $\{\mu_s^i\}_{i\in\{1,\dots,1000\}}$ les 1000 valeurs de cette erreur initiale sur les états $\boldsymbol{v}$ et on définit $\left\{b_i^{algo}, i \leq 1000, algo \in \{exact, Gauss, mixed\}\right\}$ où b_i^{algo} vaut -1 si l'algorithme $algo$ converge vers le même minimum avec l'erreur d'initialisation de μ_s^i et sans erreur et $+1$ s'il converge vers un autre minimum. On dit que ces deux minima sont les mêmes si l'erreur μ_s entre eux est inférieure à 0,01.

Pour chacun des trois algorithmes, nous définissons ensuite le taux de divergence $\delta_{algo}(\mu_s)$ associé à une erreur μ_s donnée. Il prend en compte tous les b_i^{algo} mais donne plus de poids à ceux dont l'indice i est tel quel μ_i est proche de μ_s :

$$\delta_{algo}(\mu_s) = \frac{1}{2}\Big(1 + \frac{\sum_{i=1}^{1000} b_i^{algo} e^{-\left(\frac{\mu_s-\mu_i}{\alpha}\right)^2}}{\sum_{i=1}^{1000} e^{-\left(\frac{\mu_s-\mu_i}{\alpha}\right)^2}}\Big). \tag{2.26}$$

En d'autres termes, si la majorité des μ_i au voisinage de μ_s sont associés à b_i^{algo} égal à -1 (c'est-à-dire que l'algorithme converge vers le bon minimum avec des erreurs d'initialisation autour de μ_s), alors $\delta_{algo}(\mu_s)$ sera proche de 0. Si les b_i valent 1 (c'est-à-dire que l'algorithme ne converge pas vers le minimum approprié), alors, $\delta_{algo}(\mu_s)$ sera proche de 1. Le paramètre α de (2.26) quantifie à quelle distance de μ_s on "sort du voisinage à étudier". Pour les résultats que nous recherchons, nous avons choisi $\alpha = 0,1$. La figure 2.6 montre les taux de divergence des trois algorithmes dans les quatre configurations décrites dans la section 2.5.1 avec 4 ou 15 types de mesures et 5000 ou 500 000 mesures au total.

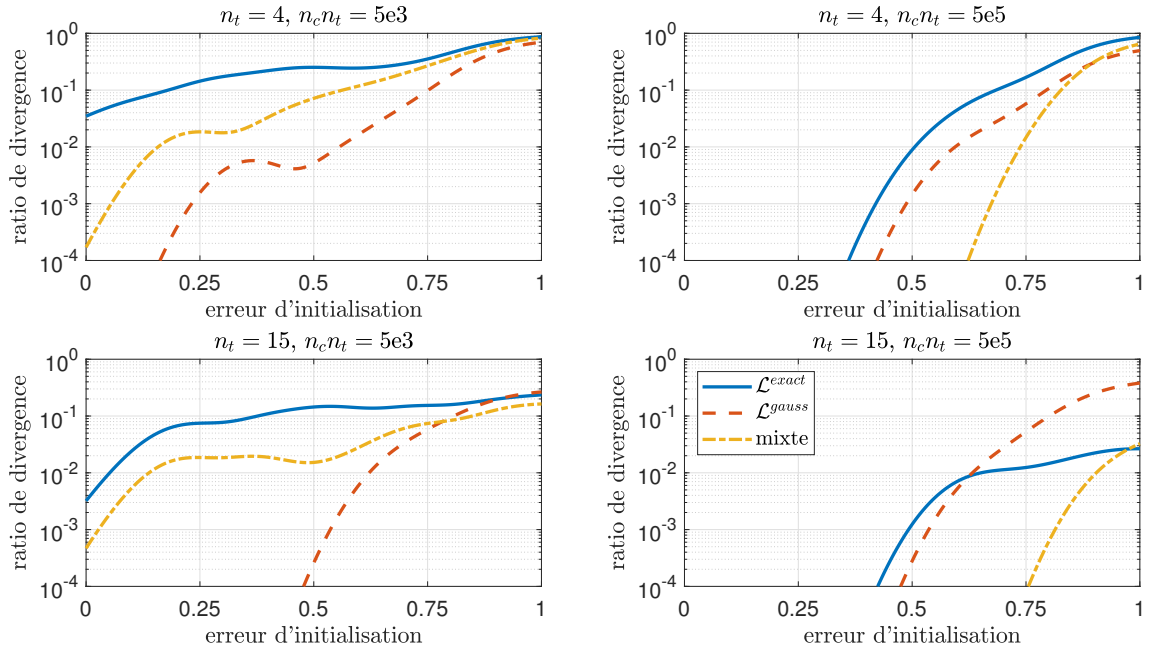


FIGURE 2.6 : Convergence des différents algorithmes de maximum de vraisemblance avec différentes erreurs d'initialisation. Les quatre configurations des quatre sous-plots sont les mêmes que celles de la figure 2.4.

Les deux graphiques de droite ne présentent qu'un intérêt limité pour nous, car le taux de divergence est toujours très faible ($\leq 10^{-2}$) pour les erreurs inférieures à 0,5. Or pour $n_c n_t = 5 \times 10^5$ (ce qui est le cas pour les deux graphiques de droite), d'après la figure 2.4, l'algorithme récursif donne toujours une estimation qui correspond à une erreur très inférieure à 0,5 et PhaseCut le fait également assez rapidement (après plus de 5000 itérations). Pour ces erreurs (dans les deux graphiques de gauche), le meilleur algorithme semble être la minimisation de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$. Ce n'est pas surprenant, la robustesse à l'erreur d'initialisation est la raison pour laquelle nous avons introduit $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$. L'algorithme mixte n'atteint pas tout à fait la même robustesse, mais il constitue certainement une amélioration par rapport à l'algorithme qui minimise $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$, qui présente les pires performances pour les erreurs d'initialisation pertinentes. Il convient de noter que le nom donné à δ "taux de divergence" est un peu sévère, l'algorithme de maximisation de vraisemblance ne diverge jamais en pratique, et il peut converger vers un minimum local qui est techniquement faux, mais assez proche du vrai minimum. δ n'est cependant pas inutile, et la figure 2.6 nous montre que, même avec ce critère sévère, l'algorithme mixte ou l'algorithme qui minimise $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ converge vers le vrai minimum si l'erreur d'initialisation est inférieure à 0,5 . D'après la figure 2.4, 5000 itérations de PhaseCut et de l'algorithme récursif produisent généralement une erreur inférieure à 0,5. Par conséquent, nous choisissons d'utiliser l'algorithme récursif lorsque cela est possible, c'est-à-dire avec la configuration de la section 2.3 avec 15 types de mesures pour 7 qubits (parce qu'il est plus rapide que PhaseCut) et lorsque PhaseCut doit être utilisé (donc avec 4 types de mesures), nous n'effectuons que 5 000 itérations. Nous pourrions laisser PhaseCut fonctionner plus longtemps, mais notre implémentation de l'algorithme ML est plus rapide.

2.5.4 Combinaison des algorithmes d'initialisation avec les algorithmes de maximum de vraisemblance sur 7 qubits

Cette section vise à tester les algorithmes des sections 2.2 et 2.3, affinés avec les trois algorithmes de la section 3.3.1 sur $n_{qb} = 7$ qubits, avec les quatre configurations décrites dans la section 2.5.1. Pour chaque configuration et pour chaque version de l'algorithme ML, quatre estimations de $\boldsymbol{v}$ sont calculées :

- L'estimation initiale, donc $\widehat{\boldsymbol{v}}_{pc}$ pour la configuration avec 4 types de mesures ou $\widehat{\boldsymbol{v}}_{rec}$ pour la configuration avec 15 types de mesures. Elle ne dépend pas du choix de l'algorithme ML.
- $\widehat{\boldsymbol{v}}_{ml}$, le résultat de l'optimisation de la vraisemblance (minimisant $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ ou $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ ou les deux successivement) initialisée à l'estimation initiale.
- $\widehat{\boldsymbol{v}}_{ref}$, le résultat de l'optimisation de la vraisemblance initialisée à la vraie valeur de $\boldsymbol{v}$ (non disponible dans la pratique, il devrait s'agir du maximum global de vraisemblance : si $\widehat{\boldsymbol{v}}_{ml} = \widehat{\boldsymbol{v}}_{ref}$ alors l'estimation initiale était suffisamment bonne). Nous appelons $\widehat{\boldsymbol{v}}_{ref}$ la référence, qui a déjà été définie (mais pas nommée) dans la section 2.5.2 et représentée à la figure 2.5.
- Et $\widehat{\boldsymbol{v}}_{rnd}$ qui est le résultat de l'optimisation de la vraisemblance initialisée à un vecteur normalisé généré aléatoirement (si $\widehat{\boldsymbol{v}}_{rnd}$ n'est pas pire que $\widehat{\boldsymbol{v}}_{ml}$ alors l'estimation initiale n'était pas nécessaire et on ne peut utiliser que l'algorithme du maximum de vraisemblance initialisé de manière aléatoire).

Pour chaque configuration, 1000 états $\boldsymbol{v}$ à estimer sont générés aléatoirement (avec la même méthode que celle utilisée pour générer les 50 états initiaux de la section 2.5.1). Nous calculons les

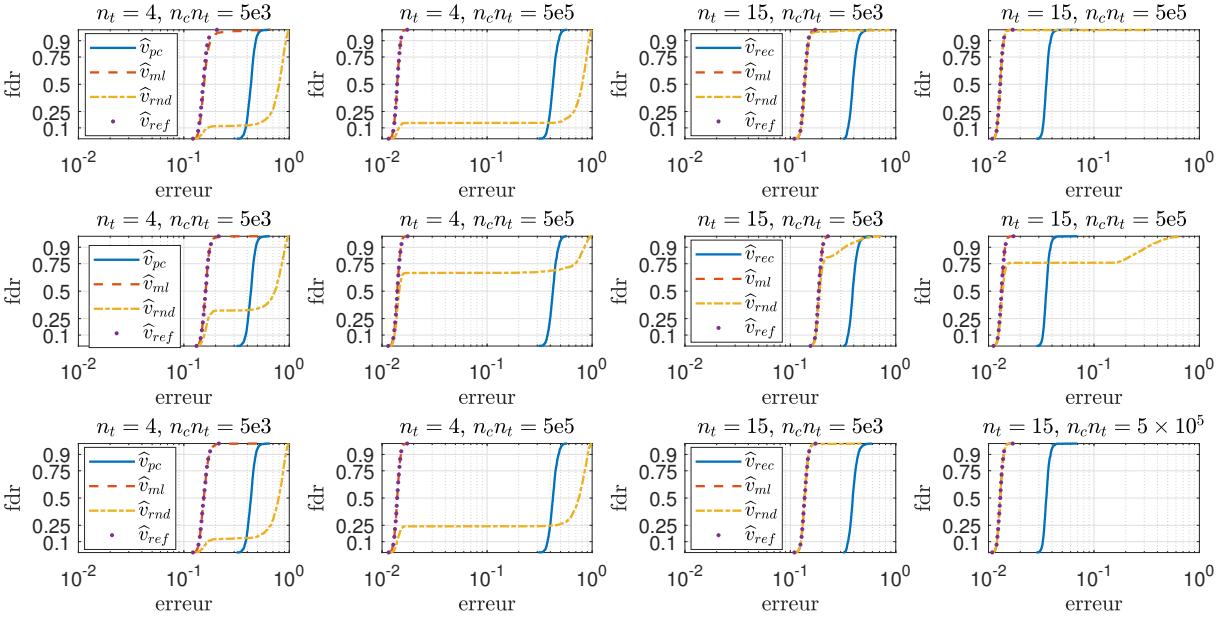


FIGURE 2.7 : fdr empirique de l'erreur de QST. Dans les quatre graphes de la ligne du haut, $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ est minimisé. Pour la ligne du milieu, il s'agit de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$. La dernière ligne correspond à l'algorithme mixte. La courbe continue bleue (la courbe la plus à droite dans tous les graphiques) est la fdr de l'erreur de l'algorithme d'initialisation (PhaseCut pour 4 types de mesures et l'algorithme récursif pour 15 types de mesures), elle ne dépend pas de l'algorithme de maximisation de la vraisemblance et est la même sur chaque ligne. Les légendes des deuxième et quatrième colonnes ne sont pas affichées afin de garder les courbes visibles, elles seraient les mêmes que les légendes des première et troisième colonnes respectivement.

estimations de chaque $\boldsymbol{v}$ avec les différents algorithmes et affichons la fdr empirique des erreurs dans la figure 2.7 (chaque rangée de graphiques correspond à un algorithme ML différent).

Les performances des trois algorithmes de ML sont assez similaires (si l'on exclut l'initialisation aléatoire), mais certaines différences peuvent être notées :

- L'algorithme qui minimise $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ est supposé être moins robuste à l'erreur d'initialisation que les autres. Cela n'est apparent que pour la configuration avec types de 4 mesures et $n_c n_t = 5000$. $\widehat{\boldsymbol{v}}_{ml}$ n'est pas aussi précis que $\widehat{\boldsymbol{v}}_{ref}$.

- L'algorithme qui minimise $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ n'a pas ce problème, $\widehat{\boldsymbol{v}}_{ml}$ et $\widehat{\boldsymbol{v}}_{ref}$ sont toujours indiscernables. Cependant, la version de $\widehat{\boldsymbol{v}}_{ref}$ calculée en minimisant $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ n'est pas aussi précise que la version qui minimise $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$. On peut le constater en comparant les deux premières lignes de la figure 2.7, mais c'est plus visible sur la figure 2.5 qui représente les performances des trois références dans un seul graphique.

- L'algorithme mixte semble combiner les avantages de ceux basés sur $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ et $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$. $\widehat{\boldsymbol{v}}_{ml}$ est presque égal à $\widehat{\boldsymbol{v}}_{ref}$et $\widehat{\boldsymbol{v}}_{ref}$ est presque aussi bon avec cet algorithme mixte qu'avec $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ (voir la figure 2.5 pour une comparaison plus claire des deux valeurs de $\widehat{\boldsymbol{v}}_{ref}$).

Les performances de $\widehat{\boldsymbol{v}}_{rnd}$ (les estimateurs du maximum de vraisemblance initialisés à un point aléatoire) sont intéressantes. Avec la configuration à 4 types de mesures, elles sont toujours bien moins bonnes que $\widehat{\boldsymbol{v}}_{ml}$. Mais, avec 15 types de mesures, elles sont (presque) aussi bonnes que les estimateurs du maximum de vraisemblance initialisés à $\widehat{\boldsymbol{v}}_{rec}$ (sauf pour la minimisation

de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$). Cela peut nous amener à nous interroger sur la pertinence de l'algorithme récursif défini dans la section 2.3. Il semblerait que la structure de la matrice de mesure $\mathbf{A}_t$ est telle que l'algorithme de descente de gradient converge naturellement vers le minimum global à partir de n'importe quel point initial. Cependant, l'algorithme récursif reste utile, car il est très rapide et accélère la maximisation de la vraisemblance (voir tableau 2).

Nous pouvons également comparer les performances des deux algorithmes d'initialisation $\widehat{\boldsymbol{v}}_{pc}$ ou $\widehat{\boldsymbol{v}}_{rec}$ (courbe bleue continue) avec $\widehat{\boldsymbol{v}}_{ml}$ (courbe rouge en pointillés). L'erreur sur $\widehat{\boldsymbol{v}}_{ml}$ est au moins 3 fois plus petite (ou encore plus petite pour $\widehat{\boldsymbol{v}}_{pc}$ et $n_c n_t = 500000$) que celle des algorithmes d'initialisation. Cela montre que la maximisation de la vraisemblance est très utile pour réduire l'erreur. Comparer la précision de l'algorithme d'initialisation avec $\widehat{\boldsymbol{v}}_{rnd}$ n'est pas judicieux, car $\widehat{\boldsymbol{v}}_{pc}$ et $\widehat{\boldsymbol{v}}_{rec}$ peuvent être améliorés avec l'algorithme ML alors que $\widehat{\boldsymbol{v}}_{rnd}$ ne peut pas l'être, car il s'agit d'un minimum local de la vraisemblance. En outre, avec $n_c n_t = 5000$, $\widehat{\boldsymbol{v}}_{pc}$ et $\widehat{\boldsymbol{v}}_{rec}$ ont une précision similaire (respectivement pour 4 et 15 types de mesures). Et, avec $n_c n_t = 500000$, $\widehat{\boldsymbol{v}}_{rec}$ est une bien meilleure estimation que $\widehat{\boldsymbol{v}}_{pc}$ car l'algorithme PhaseCut est limité à 5000 itérations (lui permettre suffisamment d'itérations pour converger correctement serait beaucoup plus lent et moins précis que la maximisation de la vraisemblance).

Après l'optimisation de la vraisemblance, les performances de $\widehat{\boldsymbol{v}}_{ml}$ avec 15 et 4 types de mesures sont comparables (la configuration à 15 mesures est légèrement meilleure). En outre, l'erreur finale est environ 10 fois moins importante lorsque le nombre de mesures est multiplié par 100. Cela signifie que pour plus de 5000 mesures, on peut sans doute extrapoler l'erreur (et donc sa fdr), puisque l'erreur est proportionnelle à $n_c n_t^{-1/2}$.

Le fait qu'il existe un ensemble de mesure nulle sur lequel la récupération de phase est impossible avec l'algorithme récursif utilisé pour calculer $\widehat{\boldsymbol{v}}_{rec}$ (voir section 2.3) s'avère ne pas être un problème. Nous aurions pu nous attendre à voir des valeurs aberrantes dans l'erreur de $\widehat{\boldsymbol{v}}_{rec}$ et les $\widehat{\boldsymbol{v}}_{ml}$ calculées à partir de cette erreur si les $\boldsymbol{v}$ générés aléatoirement étaient suffisamment proches de l'ensemble en question. Ce n'est pas le cas : chacun des 1000 états initiaux a été récupéré avec succès avec une erreur raisonnable. Il en va de même lors de l'utilisation de PhaseCut avec la configuration à 4 types de mesures. Même si nous n'avons pas pu prouver l'injectivité, la QST fonctionne bien dans la pratique et il n'y a pas de valeurs aberrantes dans l'erreur si les algorithmes appropriés sont utilisés.

Les tableaux 2.5.4 et 2.5.4 indiquent le temps d'exécution médian de tous les algorithmes sur un coeur Intel Xeon Gold 6226R 2,9 GHz. Tous les scripts ont été exécutés sur un seul "thread" de Matlab. Il n'y a pas de différences significatives entre les trois algorithmes ML lorsqu'ils ne sont pas initialisés au hasard. L'initialisation aléatoire n'est jamais pertinente, car (i) pour la configuration à 4 types de mesures, elle est relativement rapide (car elle nous épargne l'étape d'initialisation avec PhaseCut) mais la perte de précision est catastrophique (courbes en pointillés oranges sur les graphes de gauche de la figure 2.7) ; et (ii) pour la configuration à 15 types de mesures, elle est toujours plus lente (parfois beaucoup plus lente) que la maximisation de la vraisemblance avec une initialisation correcte.

En conclusion, si la seule erreur sur les mesures est l'"erreur multinomiale" (erreur due au nombre fini de mesures), nous recommandons d'utiliser l'algorithme mixte pour la vraisemblance, car il s'agit d'un bon compromis entre la minimisation de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ et la minimisation de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$. Le choix entre 4 et $2n_{qb}+1$ types de mesures est moins évident. La première est évidemment plus simple pour l'opérateur et l'optimisation de la vraisemblance est plus rapide (voir tableau 1 et tableau 2), mais :

- Elle donne un résultat légèrement moins précis. L'erreur médiane avec l'algorithme mixte et $n_c n_t = 5000$ est de $0,22$ contre $0,19$ avec 15 types de mesures.

- Nous n'avons pas d'algorithme explicite permettant de récupérer l'état à partir des mesures

en un nombre connu d'itérations. Nous devons utiliser sur PhaseCut qui n'est pas précis. PhaseCut est également plus lent que l'algorithme récursif, mais le temps gagné pendant l'algorithme ML mixte compense largement la lenteur de PhaseCut (voir les tableaux 1 et 2).

- Nous avons expliqué (dans la section 2.2.2) pourquoi nous pensons que les mesures sont injectives, et dans la pratique, les 1000 états testés ont été récupérés, mais nous n'avons pas pu prouver l'injectivité.

	$n_c n_t = 5000$	$n_c n_t = 500000$
algorithme récursif	0,17 s	0,17 s
minim. de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. à $\widehat{\boldsymbol{v}}_{rec}$	44.4 s	10.9 s
minim. de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. aléatoire	272 s	94.4 s
minim. de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. à $\widehat{\boldsymbol{v}}_{rec}$	38.8 s	16.7 s
minim. de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. aléatoire	84.9 s	126.7 s
algo. mixte init. à $\widehat{\boldsymbol{v}}_{rec}$	47.8 s	26.1 s
algo. mixte, init. aléatoire	62 s	38.4 s

TABLE 2.1 : Temps moyen d'exécution pour les configurations avec 15 types de mesures.

	$n_c n_t = 5000$	$n_c n_t = 500000$
PhaseCut	16.9 s	17.4 s
minim. de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. à $\widehat{\boldsymbol{v}}_{pc}$	11.4 s	8.3 s
minim. de $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. aléatoire	22 s	24.7 s
minim. de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. à $\widehat{\boldsymbol{v}}_{pc}$	8.1 s	5.2 s
minim. de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ init. aléatoire	16.6 s	21.3 s
algo. mixte init. à $\widehat{\boldsymbol{v}}_{pc}$	6.8 s	4.7 s
algo. mixte init. aléatoire	10.6 s	12.8 s

TABLE 2.2 : Temps moyen d'exécution pour les configurations avec 4 types de mesures.

2.5.5 États mélange

Dans la présente section, nous testons nos algorithmes sur des états mélange. Tous nos algorithmes sont conçus pour des états purs et renvoient des estimations d'états purs. Nous ne changerons pas cela mais nous utiliserons des états mélange qui sont proches d'un état pur pour générer les mesures. Nous pouvons ainsi tester les performances de nos algorithmes si l'état considéré n'est pas tout à fait pur. Les états générés sont de rang 5 (arbitraire), la matrice densité qui représente l'état mixte est la suivante

$$\boldsymbol{\rho} = p_0 \boldsymbol{v}_0 \boldsymbol{v}_0^* + \sum_{k=1}^{4} p_k \boldsymbol{v}_k \boldsymbol{v}_k^* \tag{2.27}$$

où p_0 est la valeur propre la plus élevée de $\boldsymbol{\rho}$. Plus p_0 est élevée, plus l'état considéré est proche de la pureté. $\boldsymbol{v}_0$ est l'état associé, et $\boldsymbol{v}_1$, ..., $\boldsymbol{v}_4$ sont choisis de telle sorte que $\boldsymbol{v}_0$, ..., $\boldsymbol{v}_4$ soient tous orthogonaux les uns par rapport aux autres. Si nous voulions approximer $\boldsymbol{\rho}$ avec un état

pur, $\boldsymbol{v}_0$ serait la meilleure approximation (au sens de la fidélité de la littérature 1.1.9). Nous jugerons les performances de nos estimateurs en fonction de leur proximité avec $\boldsymbol{v}_0$. L'erreur vaut $\mu_s = ||\boldsymbol{v}_0 - \widehat{\boldsymbol{v}}.e^{-i\xi}||_2$ avec ξ qui minimise la métrique, comme avec (2.25). Avec cette définition, le lien entre μ_s et la fidélité f établi dans la section 2.5.1 ne s'applique plus ($f \neq 1 - \mu_s^2$).

Nous effectuons des simulations avec dix valeurs différentes de p_0 :

$$\left\{1 - 0.325 \times \left(\frac{2}{3}\right)^k\right\}_{k\in\{0,...,9\}} \tag{2.28}$$

(les valeurs numériques sont fournies en légende sur la figure 2.8). Ces valeurs sont choisies pour voir dans quelle mesure l'erreur varie linéairement en fonction de $1 - p_0$. La figure 2.8 affiche les fdr de l'erreur avec une échelle logarithmique pour l'axe des abscisses, et si la fdr est décalée d'un intervalle constant (en échelle logarithmique) lorsque $1 - p_0$ est multiplié par $\frac{2}{3}$ dans (2.28), alors on peut dire que la relation entre l'erreur et $1 - p_0$ est linéaire.

Pour chaque valeur de p_0, 100 vecteurs $\boldsymbol{v}_0, ..., \boldsymbol{v}_4$ sont générés aléatoirement en appliquant la transformation de Gram-Schmidt à cinq vecteurs complexes aléatoires (les parties réelle et imaginaire de chaque composante sont tirées suivant des lois gaussiennes indépendantes à variance unitaire et centrées) de dimension d. Les valeurs de $p_1, ..., p_4$ sont choisies au hasard (distribution uniforme entre 0 et 1), puis normalisées de manière à ce que $\boldsymbol{\rho}$ ait une trace unitaire. Nous continuons à simuler un nombre fini de mesures, qui créent une erreur (l'erreur multinomiale), mais nous choisissons la valeur la plus élevée de $n_c n_t$, $n_c n_t = 500000$. Nous n'utilisons dans un premier temps que l'algorithme mixte. La figure 2.8 présente la fonction de répartition empirique de l'erreur pour les deux configurations avec les dix valeurs différentes de p_0.

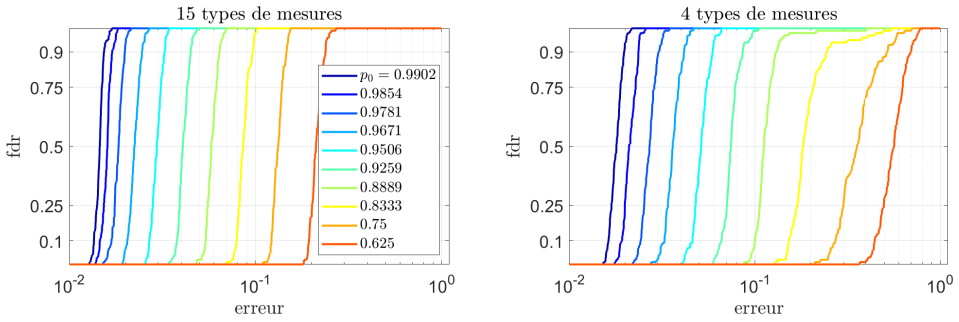


FIGURE 2.8 : fdr empirique de l'erreur de $\widehat{\boldsymbol{v}_{ML}}$ calculée avec l'algorithme mixte. Les courbes avec le p_0 le plus proche de 1 (qui correspond à état mesuré plus proche d'un état pur) sont à gauche, celles avec le plus petit p_0 sont à droite.

L'approche basée sur notre méthode originale présentée dans la section 2.3, qui utilise 15 types de mesures, est beaucoup plus résistante aux états mélange que l'approche basée sur l'algorithme PhaseCut proposé dans la littérature (voir section III), qui utilise quatre types de mesures : pour $p_0 = 0.9506$ par exemple, la médiane de l'erreur est de $0,042$ avec 15 types de mesures et de $0,072$ avec quatre types de mesures.

L'erreur semble varier assez linéairement en fonction de $1 - p_0$ sauf pour p_0 proche de 1 car, pour ces valeurs, l'erreur "normale" due au nombre fini de mesures devient plus importante. Nous observons également une non-linéarité pour la configuration 4 types de mesures : elle commence par la courbe verte ($p_0 = 0.8889$) qui a une queue légèrement plus lourde que l'autre fdr (associée à des valeurs plus petites de p_0). Elle devient plus évidente avec la courbe jaune ($p_0 = 0.8333$). Les courbes orange et rouge (les deux courbes les plus à droite) ont des formes et des pentes différentes de celles des courbes de gauche. Nous n'observons aucune de ces non-linéarités avec

la configuration de type 15 mesures, mais nous les verrions avec des p_0 plus petits (car l'erreur est bornée (≤ 1)).

On peut aussi comparer l'algorithme mixte à l'algorithme qui minimise la vraisemblance gaussienne (la vraisemblance exacte est toujours moins bonne en pratique pour des états mélange). Plutôt que d'afficher la fdr, on choisit d'afficher des boîtes à moustaches. On a pu constater (avec l'analyse de la figure 2.8) que la distribution des erreurs était à peu près toujours la même à une translation près, et les boîtes à moustaches nous permettent de condenser l'information sur la distribution, on peut donc afficher plus de distributions sur un seul graphique. La figure 2.9 représente les boîtes à moustaches des erreurs avec l'algorithme mixte et de l'algorithme qui minimise $\mathcal{L}^{gauss}$. On fait varier (en abscisse) $1-p_0$ entre les mêmes bornes (mais avec deux fois plus de points) que sur la figure 2.8.

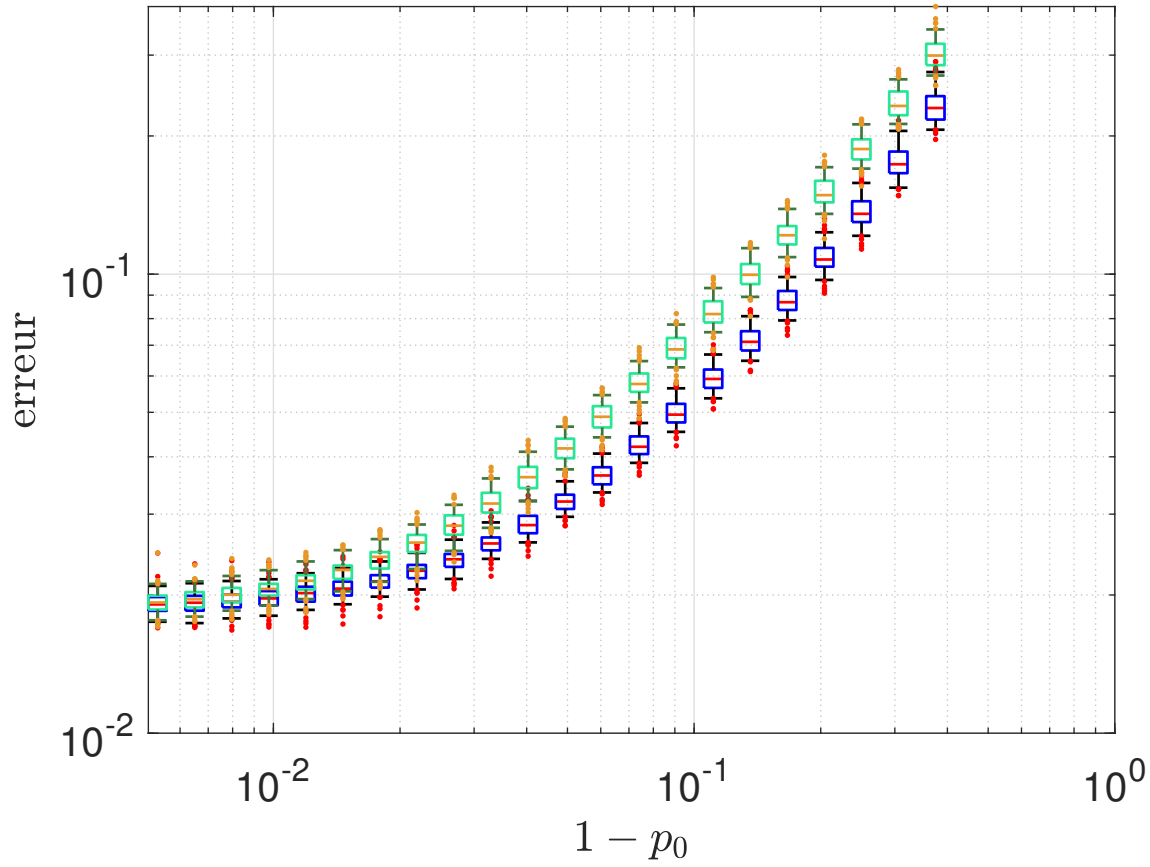


FIGURE 2.9 : Boites à moustaches des erreurs avec l'algorithme mixte (vert orange et gris) et de l'algorithme qui minimise $\mathcal{L}^{gauss}$ (en bleu, rouge et noir) pour des valeurs de $1-p_0$ allant de (à peu près) $0,005$ à $0,375$. La configuration est celle de la section 2.3 ($n_t = 15$), et on prend $n_c = 33333$. Chaque boite à moustaches représentent : la médiane de l'erreur (barre rouge ou orange au milieu), les premiers et derniers quartiles (ce sont les boites bleues ou vertes), les premiers et derniers 5-centiles (ce sont les moustaches noires ou grises), et les points en dehors des moustaches (points rouges ou oranges).

On observe la même tendance que sur la figure 2.8 : quand $1-p_0$ dépasse une certaine valeur (0.07) l'erreur dépend linéairement de $1-p_0$ (jusqu'à un certain point que nous n'observons pas avec nos valeurs de p_0). Cependant, l'observation la plus importante que nous pouvons tirer de la figure 2.9 est que l'algorithme qui minimise $\mathcal{L}^{gauss}$ a de meilleures performances que l'algorithme mixte, la différence est apparente dès $1-p_0 = 0.01$. Ce n'est pas vraiment surprenant, le modèle

multinomial qui est considéré pour calculer la vraisemblance à la fin de l'algorithme mixte n'est parfaitement adapté que si la seule source d'erreur est le fait que l'on ne fasse pas un nombre infini de mesures (l'erreur "multinomiale"). S'il existe d'autres erreurs qui ne sont pas modélisées (e.g. si $1 - p_0$ est trop élevé), alors le modèle gaussien est mieux adapté. En effet, ce dernier (i) approxime une distribution multinomiale par une distribution gaussienne, (ii) ajoute un bruit blanc gaussien au modèle de toutes les composantes de l'erreur, (iii) re-normalise les probabilités pour que le modèle contienne l'information que la somme des probabilités vaille 1. (ii) explique pourquoi la vraisemblance gaussienne est plus adaptée quand, en plus de l'erreur "multinomiale", il y a une petite erreur non modélisée (modéliser une erreur inconnue avec une variable aléatoire gaussienne, i.i.d. sur chaque composante est une bonne première étape).

Pour conclure, ces tests avec des états mélanges nous ont permis de constater que bien que $\mathcal{L}^{gauss}$ avait été pensée à la base pour faciliter la convergence des algorithmes de gradient. Il se trouve qu'il est très utile pour prendre en compte les erreurs non modélisées. Nous allons donc privilégier l'utilisation de $\mathcal{L}^{gauss}$ par la suite. Nous avons aussi constaté que la configuration de la section 2.3 avec $2n_{qb} + 1$ types de mesures semble être plus adaptée aux états mélange.

2.5.6 Test avec moins de 7 qubits et comparaison avec l'algorithme de Goyeneche et al.

Les sections précédentes nous ont permis de constater que (i) les algorithmes d'initialisation des sections 2.2 (phaseCut, 4 types de mesures) et 2.3 (algorithme récursif, $2n_{qb}+1$ types de mesures) avec les mesures sur lesquels ils ont été conçus sont tous les deux intéressants (nous avons une préférence pour le premier) (ii) ils doivent être améliorés avec un algorithme de maximum de vraisemblance, nous préférons l'algorithme gaussien car il simplifie l'optimisation et améliore les performances si des imperfections du systèmes n'ont pas été minimisées (si l'état mesuré est un état mélange par exemple).

Dans la présente section, nous allons tester les deux algorithmes d'initialisation améliorés avec l'algorithme qui minimise $\mathcal{L}^{gauss}$. Nous allons aussi les comparer avec l'algorithme de [GCE+15] sur les mesures adaptées, que l'on peut aussi améliorer avec le même algorithme de maximum de vraisemblance. Nous faisons varier le nombre de qubits n_{qb} de 1 à 7 (pour un seul qubit, l'algorithme de Goyeneche et al. [GCE+15] et l'algorithme de la section 2.2 ne sont pas définis). Pour chaque nombre de qubits 1000 états aléatoires sont crées, ils sont mesurés avec les trois ensembles de types de mesures associés aux 3 algorithmes d'initialisation. L'algorithme de la section 2.2 (phaseCut) ainsi que l'algorithme de [GCE+15] sont définis avec $n_t = 4$ types de mesures, alors que l'algorithme de la section 2.3 est défini avec $n_t = 2n_{qb} + 1$ types de mesures. Comme dans les sections précédents, nous allons comparer ces différents algorithmes en gardant le nombre total de mesures ($n_c n_t$) constant à $n_c n_t = 5000$ pour tous les algorithmes et tous les nombres de qubits. La figure 2.10 représente les boîtes à moustaches des erreurs de QST pour les 3 algorithmes avant et après la maximisation de la vraisemblance gaussienne.

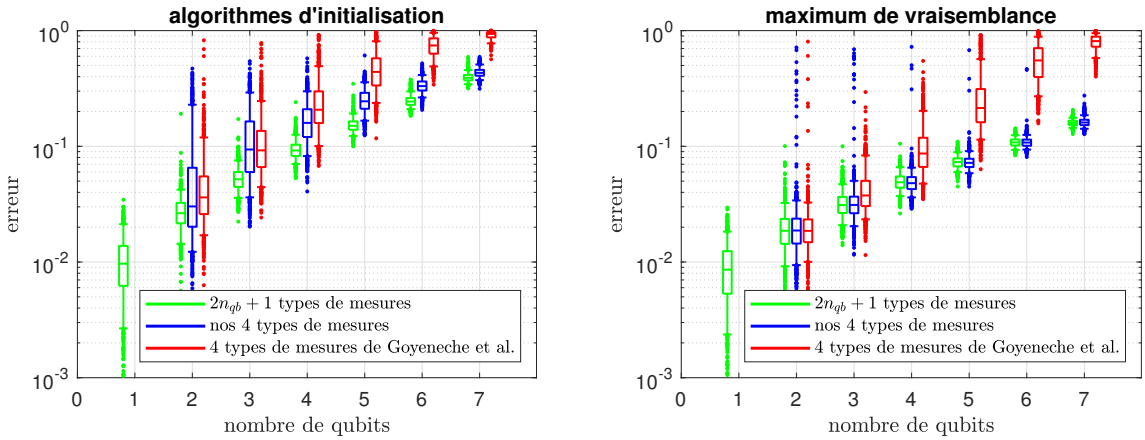


FIGURE 2.10 : Boîtes à moustaches des erreurs de QST quand le nombre de qubits varie. Le graphe de gauche représente les erreurs des algorithmes d'optimisation, de l'algorithme récursif de la section 2.3 (en vert), de l'algorithme récursif de la section 2.2 (en bleu), de l'algorithme de Goyeneche et al. [GCE+15] (en rouge). Tous ces algorithmes sont implémentés avec les types de mesures pour lesquels ils ont été conçus (voir légende) avec un nombre de mesures total $n_c n_t = 5000$ constant sur toutes les boîtes. Le graphe de droite représente les performances de l'algorithme de maximum de vraisemblance avec la vraisemblance gaussienne avec les trois configurations des algorithmes d'initialisation du graphe de gauche (configuration de la section 2.2, configuration de la section 2.3 et configuration de [GCE+15]). Chaque boite à moustache permet de lire la médiane, les premier et dernier quartiles, les premier et derniers 5-centiles et les "outliers" comme pour la figure 2.9.

Les résultats de la méthode de [GCE+15] nous ont surpris par leur médiocrité avec quatre qubits et plus. L'erreur devient bien pire dans la configuration de [GCE+15] qu'avec les deux autres quand le nombre de qubits augmente. Cela pourrait être dû au fait que nous avons mal implémenté la méthode, mais :

- l'algorithme de [GCE+15] (non-amélioré par le maximum de vraisemblance gaussien) calcule la j-ème composante du vecteur $\boldsymbol{v}$ en faisant un produit de j complexes, chacun de ces complexes est défini à partir des probabilités mesurées. Plus j est grand moins l'estimée sera précise, il est donc logique que la précision diminue quand la taille du vecteur d'état augmente (donc un n_{qb} augmente).

- Même si nous avions mal implémenté l'algorithme de [GCE+15], nous ne nous en servons que pour l'initialisation, l'algorithme de maximum de vraisemblance est presque le même pour la configuration de [GCE+15] que pour les deux autres (seule la matrice $\mathbf{A}$ change), et les performances après maximisation de la vraisemblance sont particulièrement mauvaises dans la configuration de [GCE+15]. Ce n'est pas dû au fait que l'initialisation fait converger vers le mauvais minimum (nous avons fait les tests en initialisant l'état sur le vrai $\boldsymbol{v}$ pour nous en assurer), le problème est que la vraisemblance est mal conditionnée et est très sensible au bruit

Il convient de noter que les performances de l'algorithme de [GCE+15] sont mauvaises malgré le fait que nous n'ayons pas modélisé le fait que les mesures qu'il propose de réaliser sont plus complexes à réaliser en pratique et introduisent sans doute des erreurs supplémentaires. Les performances de nos deux algorithmes (phaseCut puis ML sur les 4 types de mesures de la section 2.2 et algorithme récursif puis ML sur les 4 types de mesures de la section 2.3) sont similaires. Comme on peut s'y attendre, les performances se dégradent quand le nombre de qubits

augmente. En effet, le nombre de paramètres dépend linéairement de la dimension $d = 2^{n_{qb}}$, le nombre de probabilités estimées dépend aussi linéairement de la dimension, et le ratio entre les deux reste constant, cependant, la valeur des probabilités que l'on estime est de l'ordre de $1/d$ (les probabilités somment à 1), comme on garde le nombre de mesure total constant, il est donc logique que l'on estime ces probabilités de plus en plus faibles avec une erreur relative qui augmente, cela dégrade la précision de l'estimation de $\boldsymbol{v}$.

2.6 Conclusion

Au début de la thèse, nous avions l'intention d'utiliser un algorithme de QST ("Quantum State Tomography" tomographie d'état) de la littérature. Nous avions deux contraintes : l'algorithme devait être (i) adapté aux états purs, et (ii) il devait n'utiliser que des mesures non intriquées à d résultats possibles. Nous n'avons trouvé aucun algorithme de la littérature qui remplisse ces deux conditions. Nous avons donc défini deux ensembles (originaux) de types de mesures pour la QST des états purs. Les 4 types de mesures définies dans la section 2.2, et les $2n_{qb} + 1$ de la section 2.3. Nous avons aussi défini des algorithmes adaptés (un original et un issu de la littérature). Et nous proposons d'améliorer le résultat en trouvant l'état qui maximise la vraisemblance des mesures. Nous ne sommes pas les premiers à proposer cette idée, la log-vraisemblance négative des mesures quantiques $\mathcal{L}^{exact}$ est connue depuis longtemps [HŘFJ04] (au moins [JFH03]), mais avec notre méthode d'optimisation adaptée aux états purs, une version gaussienne régularisée de la log-vraisemblance négative $\mathcal{L}^{gauss}$ est plus facile à minimiser. Nous proposons aussi un algorithme mixte qui minimise $\mathcal{L}^{gauss}$ puis $\mathcal{L}^{exact}$.

Ces algorithmes peuvent être combinés de plusieurs façons et nous avons un choix à faire. Dans les chapitres suivants, nous voulons utiliser des mesures par défaut et un algorithme par défaut pour la QST. **Par défaut, les types de mesures que l'on considère pour la QST sont les $2n_{qb}+1$ types de mesures de la section 2.3, et l'algorithme que l'on utilise est l'algorithme récursif de la section 2.3.2 dont la sortie sert d'entrée à l'algorithme de maximum de vraisemblance qui minimise $\mathcal{L}^{gauss}$.** Ce choix est discutable, et, pour le faire, nous sommes partis du principe que (i) il y a peut-être d'autres sources d'erreur que l'erreur "multinomiale", et elles peuvent être prises en compte par le modèle gaussien régularisé et que (ii) faire plus de types de mesures tout en gardant le nombre de mesures total constant (augmenter n_t mais garder $n_t n_c$ constant) n'augmente pas la complexité d'implémentation de manière significative.

Chapitre 3

Tomographie de processus

Sommaire

Dans la section 3.1 nous présentons un premier algorithme de QPT (quantum process tomography) qui repose sur des estimations des états mesurés. Ces estimations sont calculées à l'aide d'un algorithme de QST. Dans un premier temps, nous ne nous soucions pas de savoir quel algorithme de QST du chapitre précédent peut être utilisé, nous supposons juste que des copies des états ont été mesurées et que ces mesures nous donnent des estimation des états. Nous avons décrit cet algorithme dans [VD23b] accepté dans Physical Review A en décembre 2023.

Dans la section 3.2 nous étudions les situations dans lesquelles l'algorithme de la section 3.1 permet d'estimer la matrice $\mathbf{M}$ qui définit le processus sans erreur (si on ne prend pas en compte les erreurs de QST). Nous cherchons ensuite une condition nécessaire et suffisante pour que la QPT soit techniquement possible (i.e. tous les processus distincts donnent des mesures distinctes) et nous montrons que notre algorithme de QPT fonctionne toujours quand cette condition est satisfaite. Les résultats de cette section sont aussi dans [VD23b].

Finalement, dans la section 3.3 nous présentons un deuxième algorithme qui utilise les mesures directement (sans utiliser l'estimée des états sur lesquels elles ont été faites) et qui est plus précis si on prend en compte les erreurs sur les mesures et les erreurs de QST, mais il doit

être initialisé à l'estimée du premier algorithme. Cet algorithme a été présenté de (façon moins détaillée) dans [VDD21] et [VD22].

3.1 Tomographie de processus à partir des résultats de la QST

3.1.1 Dispositif de QPT

Notre algorithme de QPT est conçu pour résister aux erreurs systématiques sur les états initiaux. Pour ce faire, nous supposons que les états initiaux $(\boldsymbol{v}_1, ..., \boldsymbol{v}_{n_i})$ sont inconnus mais que les mesures sont effectuées à différents instants, voir la figure 3.1.

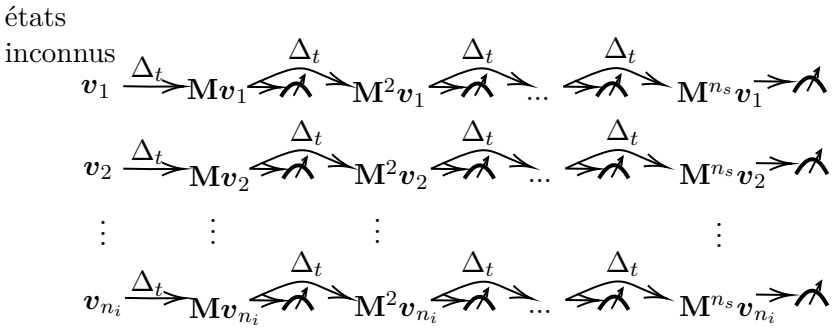


FIGURE 3.1 : Dispositif de QPT semi-aveugle, état initiaux non mesurés. Les doubles flèches signifient que $n_c n_t$ états sont mesurés, ils ne seront plus utilisés (flèche droite), et que le processus unitaire à identifier est appliqué aux autres états (flèche courbée).

Le nombre d'états initiaux est de n_i. Ils sont mesurés après avoir attendu Δ_t, $2\Delta_t$, ... ou $n_s\Delta_t$ (n_s est le nombre maximum de délais), et, d'un instant de mesure à l'autre, l'état est multiplié par la matrice $\mathbf{M}$ associée au processus. Le nombre de types de mesures effectuées sur chaque valeur d'état mesuré est appelé n_t, il dépend de l'algorithme de QST ($n_t = 2n_{qb}+1$ pour l'algorithme de la section 2.3 et $n_t = 4$ pour l'algorithme de la section 2.2). Comme pour la QST, nous choisissons d'effectuer chaque type de mesure n_c fois afin d'estimer les probabilités de chaque résultat. Chaque état d'entrée ne peut être mesuré qu'une seule fois, nous avons donc besoin de $n_s n_t n_c$ copies de chacun des n_i états d'entrée, pour un total de $n_i n_s n_t n_c$ états d'entrée préparés.

On suppose que les états initiaux sont préparés avec des portes mono-qubit, car cela minimise les risques de décohérence ([KBGK18]). Dans les sections 3.3.1, nous verrons que nos algorithmes de QPT par maximisation de la vraisemblance sont plus simples et plus efficaces si les états d'entrée sont non intriqués. Or des états d'entrée préparés avec des portes mono-qubit sont non intriqués si l'on considère que les portes mono-qubit sont bien unitaires.

Les états initiaux $(\boldsymbol{v}_1, ..., \boldsymbol{v}_{n_i})$ ne sont jamais mesurés directement. Nous pourrions imaginer une configuration similaire dans laquelle les états initiaux seraient mesurés et un délai de moins serait pris en compte. Cette configuration est représentée sur la figure 3.2. Les deux dispositifs sont presque équivalents, on passe de la figure 3.2 à la figure 3.1 en changeant les états initiaux $\boldsymbol{v}_j$ en $\mathbf{M}\boldsymbol{v}_j$ (la seule différence est que, dans le cas de la figure 3.1, les états non intriqué à considérer ne sont pas directement ceux que l'on mesure, mais leurs antécédents par $\mathbf{M}$), et nous verrons que les conditions sur les états initiaux pour que $\mathbf{M}$ soit identifiable sont stables (i.e. elles restent satisfaites) par multiplication par une matrice unitaire. Donc, les conditions sur les $\boldsymbol{v}_j$ de la figure 3.2 pour que $\mathbf{M}$ soit identifiable sont donc les mêmes que les conditions sur les $\boldsymbol{v}_j$ de la figure 3.1 pour que $\mathbf{M}$ soit identifiable, et, en pratique, les performances sont

très similaires. Nous choisissons de n'étudier que le circuit de la figure 3.1 dans un premier temps. Nous avons fait ce choix car certaines implémentations des ordinateurs quantiques ne nous permettent pas de mesurer certains états juste après les avoir préparé (voir la légende de la figure 5.15 dans la section 5.2). Les configurations des figures 3.1 et 3.2 sont beaucoup plus

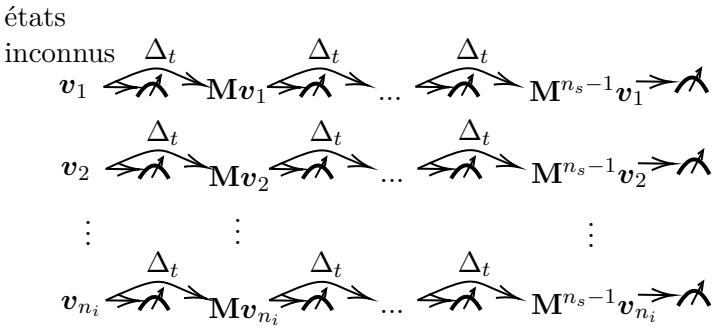


FIGURE 3.2 : Dispositif alternatif avec états initiaux mesurés

proches de la configuration de la SQPT (comme dans [CN97] décrit en section 1.4) qu'il n'y paraît à première vue. En fait, si $n_s = 2$ (nous allons étudier ce cas en détail) et $n_i = d^2$ (nous n'étudierons jamais de si grandes valeurs de n_i avec des processus unitaires), la seule différence entre le circuit de la figure 3.1 et la configuration de la SQPT, est que les $\{\mathbf{M}\boldsymbol{v}_k\}_k$ (qui jouent le même rôle que les états d'entrée pour la SQPT) ne sont pas fixés à des valeurs prédéterminées. Ils sont cependant connus, car nous effectuons plusieurs types de mesures sur des copies de ces états, ce qui nous permet de réaliser la QST. Dans l'autre cas extrême (du point de vue de n_s et n_i), avec un seul état initial $n_i = 1$, nous pouvons encore rendre la QPT possible en effectuant des mesures sur des copies de l'état après qu'elles ont traversé le processus une ou plusieurs fois (en fonction de la copie mesurée) avec $n_s \geq d + 1$ (cette condition est nécessaire mais pas suffisante). Ce cas extrême est le premier que nous avons étudié (dans [VDD21] avec $n_{qb} = 2$). Il est plus simple à réaliser en pratique, car une seule valeur d'état initial est préparée.

Les résultats de l'expérience sont les nombres d'occurrences de chaque résultat de chaque mesure. Pour chacun des $n_i \times n_s$ états mesurés avec chacun des n_t types de mesures, nous comptons le nombre de fois que chacun des d résultats s'est produit. La somme de ces d comptes est égal à n_c, et chacun de ces groupes de d mesures peut être modélisé comme une variable aléatoire qui suit une distribution multinomiale avec n_c tirages et les probabilités de chaque résultat sont déterminées par la valeur de l'état mesuré associé et par le type de mesure (voir section 2.4). Le tableau C.1 de l'Annexe C.2 montre un exemple de nombre de mesures pour une configuration de QPT donnée avec $n_i = 4$, $n_s = 2$, $n_t = 5$.

3.1.2 Idée de résolution

Nous supposons que la QST est effectuée correctement pour les états mesurés de la figure 3.1. Les états mesurés sont notés avec la convention suivante :

$$\boldsymbol{v}_{j,k} = \mathbf{M}^k \boldsymbol{v}_j, j \in \{1, ..., n_i\},\ k \in \{1, ..., n_s\} \tag{3.1}$$

et $\{\widehat{\boldsymbol{v}}_{j,k}\}_{j,k}$ sont leurs estimations. Ces estimations issues de la QST présentent chacune une indétermination de phase notée $\xi_{j,k}^{QST}$ et une erreur de QST notée $\boldsymbol{\varepsilon}_{j,k}^{QST}$

$$\widehat{\boldsymbol{v}}_{j,k} = \mathbf{M}^k \boldsymbol{v}_j . e^{i\xi_{j,k}^{QST}} + \boldsymbol{\varepsilon}_{j,k}^{QST} \quad j \in \{1, ..., n_i\},\ k \in \{1, ..., n_s\} \tag{3.2}$$

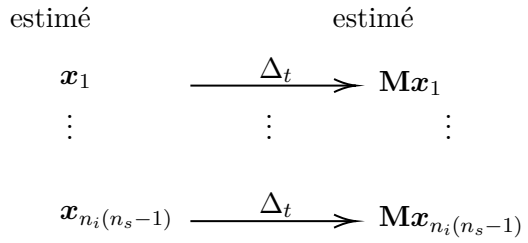


FIGURE 3.3 : Dispositif "virtuel" de QPT.

l'erreur résiduelle $\boldsymbol{\varepsilon}_{j,k}^{QST}$ est telle que $\mathbb{E}(||\boldsymbol{\varepsilon}_{j,k}^{QST}||_2) \underset{n_c \to +\infty}{\longrightarrow} 0$ ($\mathbb{E}$ est l'espérance). Lorsque nous parlons d'"erreur de QST", nous voulons dire $\boldsymbol{\varepsilon}_{j,k}^{QST}$, et non $\xi_{j,k}^{QST}$. Pour le reste de cette section, nous considérons qu'il n'y a pas d'erreur de QST : $\boldsymbol{\varepsilon}_{j,k}^{QST} = 0$ sauf indication contraire. Dans la section 1.1.2, nous avons indiqué que les phases globales des états n'ont pas d'importance. Ceci est vrai si les états sont considérés indépendamment et c'est la raison pour laquelle la QST ne peut pas récupérer la phase globale, et pourquoi nous ne considérons pas $\xi_{j,k}^{QST}$ comme une "erreur", elle affecte notre estimation même dans le cas idéal avec un nombre infini de mesures et aucune source d'erreur. Même si cette phase n'a aucune signification physique, nous allons voir qu'elle est importante pour trouver $\mathbf{M}$.

Nous savons que :

$$\boldsymbol{v}_{j,k+1} = \mathbf{M}\boldsymbol{v}_{j,k} \quad j \in \{1, ..., n_i\},\ k \in \{1, ..., n_s - 1\}. \tag{3.3}$$

Par souci de simplicité, nous définissons :

$$\begin{aligned}
\mathbf{X} &= [\boldsymbol{v}_{1,1}, ..., \boldsymbol{v}_{1,n_s-1}, \boldsymbol{v}_{2,1}, ..., \boldsymbol{v}_{n_i,n_s-1}] \\
\mathbf{Y} &= \mathbf{M}\mathbf{X} = [\boldsymbol{v}_{1,2}, ..., \boldsymbol{v}_{1,n_s}, \boldsymbol{v}_{2,1}, ..., \boldsymbol{v}_{n_i,n_s}] \\
\widehat{\mathbf{X}} &= [\widehat{\boldsymbol{v}}_{1,1}, ..., \widehat{\boldsymbol{v}}_{1,n_s-1}, \widehat{\boldsymbol{v}}_{2,1}, ..., \widehat{\boldsymbol{v}}_{n_i,n_s-1}] \\
\widehat{\mathbf{Y}} &= [\widehat{\boldsymbol{v}}_{1,2}, ..., \widehat{\boldsymbol{v}}_{1,n_s}, \widehat{\boldsymbol{v}}_{2,1}, ..., \widehat{\boldsymbol{v}}_{n_i,n_s}].
\end{aligned} \tag{3.4}$$

Avec ces notations, (3.3) devient $\mathbf{Y} = \mathbf{M}\mathbf{X}$, et nous pouvons oublier la configuration de la figure 3.1 et imaginer que nous avons affaire à la configuration plus simple de la figure 3.3 (avec la convention que $\boldsymbol{x}_\ell$ est la ℓ-ième colonne de $\mathbf{X}$).

Cette représentation est encore plus proche de la configuration de la SQPT (avec des états purs et un processus unitaire). En fait, la seule différence est que les états d'entrée ne sont pas fixés à des valeurs prédéterminées, mais sont préparés avec des portes quantiques inconnues (y compris la porte que nous essayons d'identifier). Ces états d'entrée sont connus grâce à la mesure. L'algorithme que nous allons décrire maintenant pourrait fonctionner avec la configuration de la SQPT (i.e. avec des états d'entrée prédéterminés connus $\mathbf{X}$ et des états de sortie mesurés $\mathbf{Y}$). Cependant nous considérons que les hypothèses de la SQPT sont problématiques car elles supposent entre autre que les portes qui préparent les états d'entrée (qui sont souvent des portes d'intrication sur plusieurs qubits [BKD14]) sont fixées à des valeurs prédéterminées, et la précision de la QPT sera limitée par les erreurs sur ces portes. Cela contraste avec la configuration semi-aveugle de la figure 3.1 qui remplace l'hypothèse que les états initiaux sont connus a priori par l'hypothèse qu'ils sont préparés de façon répétable et peuvent être mesurés a posteriori. Les principaux algorithmes de QPT *unitaire* [BKD14] [KLY15] ne fonctionneraient pas avec la configuration semi-aveugle parce qu'ils exigent tous que l'état d'entrée soit fixés à des valeurs prédéterminées.

Comme on a reformulé (3.3) en $\mathbf{Y} = \mathbf{M}\mathbf{X}$, et que l'on a des estimées de $\mathbf{X}$ et $\mathbf{Y}$, on pourrait penser que l'identification de la matrice $\mathbf{M}$ se résume ainsi à un simple problème d'inversion

linéaire avec une contrainte unitaire. Mais ce n'est pas le cas, car les colonnes de $\mathbf{X}$ et de $\mathbf{Y}$ ne sont connues qu'à une phase près ; et lorsque nous les considérons ensemble afin de trouver $\mathbf{M}$, leurs phases relatives (qui ne sont jamais mesurées) sont importantes[1].

Soit $\ell \in \{1, ..., n_i(n_s-1)\}$, $\exists!\ j \in \{1, ..., n_i\},\ k \in \{1, ..., n_s-1\}$ t.q. $\ell = k + (n_s-1)(j-1)$ (division euclidienne de ℓ par n_s-1), définissons la phase ξ_ℓ à partir de ces j, k :

$$\xi_\ell = \xi^{QST}_{j,k} - \xi^{QST}_{j,k+1}, \tag{3.5}$$

nous pouvons réécrire (3.3) avec les colonnes de $\widehat{\mathbf{X}}$ et $\widehat{\mathbf{Y}}$ (que nous connaissons grâce à la QST) et ces phases :

$$e^{i\xi_\ell}\widehat{\boldsymbol{y}}_\ell = \mathbf{M}\widehat{\boldsymbol{x}}_\ell \quad \forall \ell \in \{1, ..., n_i(n_s-1)\}, \tag{3.6}$$

où $\widehat{\boldsymbol{x}}_\ell$ et $\widehat{\boldsymbol{y}}_\ell$ sont les ℓ-ièmes colonnes de $\widehat{\mathbf{X}}$ et $\widehat{\mathbf{Y}}$ respectivement.

Pour tout indice ℓ_0, changer $\mathbf{M}$ en $\mathbf{M}.e^{-i\xi_{\ell_0}}$ et ξ_ℓ en $\xi_\ell - \xi_{\ell_0}\ \ \forall \ell$ ne change pas l'égalité (3.6). Par conséquent, nous pouvons également supposer qu'un ℓ_0 donné (nous expliquerons comment le choisir plus tard) est tel que $\xi_{\ell_0} = 0$ et accepter que $\mathbf{M}$ ne puisse être récupérée qu'à une phase globale près (ce qui est toujours le cas pour la QPT).

Dans la section suivante, nous expliquons comment obtenir des estimées des facteurs de phase $e^{i\widehat{\xi_\ell}}$. À partir de là, nous pouvons définir :

$$\widetilde{\boldsymbol{y}}_\ell = \widehat{\boldsymbol{y}}_\ell.e^{i\widehat{\xi_\ell}} \quad \ell \in \{1, ..., n_i(n_s-1)\}, \tag{3.7}$$

avec lesquels une estimation de $\mathbf{M}$ peut facilement être trouvée (si $n_i(n_s-1) \geq d$) en utilisant la relation linéaire :

$$\widetilde{\mathbf{Y}} = \mathbf{M}\widehat{\mathbf{X}} \tag{3.8}$$

avec

$$\widetilde{\mathbf{Y}} = [\widetilde{\boldsymbol{y}}_1, ..., \widetilde{\boldsymbol{y}}_{n_i(n_s-1)}]. \tag{3.9}$$

$\widehat{\mathbf{M}} = \widetilde{\mathbf{Y}}\widehat{\mathbf{X}}^\dagger$ fonctionne comme une solution ($.^\dagger$ est la pseudo-inverse). Mais ce n'est généralement pas une solution unitaire à cause des erreurs de QST. Le problème de trouver $\widehat{\mathbf{M}} \in \mathbb{U}_d(\mathbb{C})$ qui est la solution des moindres carrés totaux de $\widehat{\mathbf{a}_j} = \mathbf{M}\widehat{\mathbf{b}_j}\ \forall j \in \{1, ..., n\}$ a été résolu dans [Aru92] :

$$\begin{aligned} \mathbf{B} &= \widetilde{\mathbf{Y}}\widehat{\mathbf{X}}^* \\ \mathbf{U}\ \mathbf{S}\ \mathbf{V}^* &= \mathbf{B} \\ \widehat{\mathbf{M}}_{LS} &= \mathbf{U}\ \mathbf{V}^* \end{aligned} \tag{3.10}$$

où $.^*$ est le trans-conjugué, $\mathbf{U}\ \mathbf{S}\ \mathbf{V}^*$ est la décomposition en valeurs singulières de $\mathbf{B}$. Cette solution est optimale au sens des moindres carrés totaux, ce qui signifie que $\widehat{\mathbf{M}}_{LS} = (\widetilde{\mathbf{Y}} + \Delta^0_Y)(\widehat{\mathbf{X}} + \Delta^0_X)^{-1}$ où Δ^0_X, Δ^0_Y sont les solutions du problème d'optimisation suivant :

$$\{\Delta^0_X, \Delta^0_Y\} = \underset{\{\widehat{\mathbf{X}}+\Delta_X, \widetilde{\mathbf{Y}}+\Delta_Y\} \in \mathcal{L}}{\arg\min} \quad ||\Delta_X||^2 + ||\Delta_Y||^2 \tag{3.11}$$

où $\mathcal{L}$ est l'ensemble des couples de matrices de taille $d \times n_i(n_s-1)$ liées par une transformation unitaire :

[1] Par exemple, pour $n_{qb} = 1$ et avec $\mathbf{M} = \mathbf{I}_2$, si les états mesurés sont $\boldsymbol{x}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ et $\boldsymbol{x}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, on effectue la QST de $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{y}_1 = \mathbf{M}\boldsymbol{x}_1 = \boldsymbol{x}_1$ et $\boldsymbol{y}_2 = \mathbf{M}\boldsymbol{x}_2 = \boldsymbol{x}_2$, on les connaît donc à une phase près. Les résultats de la QST ne nous permettent pas d'identifier $\mathbf{M}$, même à une phase globale près. En effet, n'importe quelle matrice qui s'écrit $\mathbf{M}_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$ pour une phase $\theta \in]0, 2\pi[$ non nulle donnée, donne les mêmes états quantiques que $\mathbf{M}$ si on applique le processus correspondant aux états d'entrée représentés par $\boldsymbol{x}_1$ et $\boldsymbol{x}_2$.

$\mathcal{L} = \left\{ \mathbf{X}, \mathbf{Y} \in \mathbb{C}^{d \times n_i(n_s-1)} \text{ t.q. } \exists \mathbf{P} \in \mathbb{U}_d(\mathbb{C}) \text{ t.q. } \mathbf{Y} = \mathbf{P}\mathbf{X} \right\}$.

Nous utilisons l'approche des moindres carrés totaux parce que $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ sont tous les deux sujets à des erreurs, et Δ_X^0 et Δ_Y^0 peuvent être considérés comme notre estimation de ces erreurs. Cette approche serait optimale au sens du maximum de vraisemblance si les erreurs sur $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ étaient gaussiennes iid sur chaque composante. Ce n'est pas le cas en pratique (surtout si $n_s > 2$, dans ce cas, certaines colonnes de $\widehat{\mathbf{X}}$ sont également dans $\widehat{\mathbf{Y}}$ à une phase près, et leurs erreurs sont donc fortement corrélées), mais minimiser la norme de l'erreur n'est jamais une mauvaise idée en première approximation.

La solution $\widehat{\mathbf{M}}_{LS}$ est unique si et seulement si $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ sont toutes deux de rang plein. Ceci n'est pas explicitement énoncé dans [Aru92] mais est prouvé pour le problème de Procrustes orthogonal, dans [GVL13], auquel [Aru92] a montré qu'il est équivalent au problème des moindres carrés totaux que nous considérons.

La preuve de [Aru92] ne résout pas entièrement le problème, les auteurs montrent que la meilleure estimée de la matrice $\mathbf{P}$ qui lie $\widehat{\mathbf{X}} + \mathbf{\Delta}_X$ et $\widehat{\mathbf{Y}} + \mathbf{\Delta}_Y$ est $\widehat{\mathbf{M}}_{LS}$ de (3.10) : $\widehat{\mathbf{Y}} + \mathbf{\Delta}_Y = \widehat{\mathbf{M}}_{LS}(\widehat{\mathbf{X}} + \mathbf{\Delta}_X)$, est ceci est l'objectif de la QPT ; mais les arguments du problème de minimisation (3.11) : $\mathbf{\Delta}_X$ et $\mathbf{\Delta}_Y$ ne sont pas estimés. Ces variables représentent les erreurs sur l'estimation des états, ils ne nous intéressent pas autant que $\widehat{\mathbf{M}}_{LS}$, mais nous les utiliserons quand même dans la section 3.3.4. Dans l'Annexe B.1, nous montrons que la solution de (3.11) est :

$$\begin{array}{l} \Delta_X^0 = \frac{1}{2}\widehat{\mathbf{M}}_{LS}^* \widetilde{\mathbf{Y}} - \frac{1}{2}\widehat{\mathbf{X}} \\ \Delta_Y^0 = \frac{1}{2}\widehat{\mathbf{M}}_{LS} \widehat{\mathbf{X}} - \frac{1}{2}\widetilde{\mathbf{Y}} \end{array} \Rightarrow \begin{array}{l} \widehat{\mathbf{X}}_{LS} = \frac{1}{2}\widehat{\mathbf{M}}_{LS}^* \widetilde{\mathbf{Y}} + \frac{1}{2}\widehat{\mathbf{X}} \\ \widehat{\mathbf{Y}}_{LS} = \frac{1}{2}\widehat{\mathbf{M}}_{LS} \widehat{\mathbf{X}} + \frac{1}{2}\widetilde{\mathbf{Y}} \end{array} \tag{3.12}$$

avec le $\widehat{\mathbf{M}}_{LS}$ de (3.10). On peut facilement vérifier que $\widehat{\mathbf{Y}}_{LS} = \widehat{\mathbf{M}}_{LS}\widehat{\mathbf{X}}_{LS}$. Les états représentés par les colonnes de $\widehat{\mathbf{X}}_{LS}$ et $\widehat{\mathbf{Y}}_{LS}$ représentent les états les plus proches possibles (au sens des moindres carrés) des états estimés par la QST (et avec les colonnes re-phasées pour $\widetilde{\mathbf{Y}}$) qui sont liés par la multiplication (à gauche) par une matrice unitaire. Nous montrons aussi que les solutions de (3.11) sont uniques si et seulement si $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ sont de rang plein.

En conclusion, nous avons montré, que, si on admet qu'il est possible de calculer $\widetilde{\mathbf{Y}}$ dont les colonnes sont les mêmes que celles de $\widehat{\mathbf{Y}}$ à des phases globales près (une phase par colonne) et qui corrige les ambiguïtés de phase que l'on a besoin de lever pour la QPT, alors, au sens des moindres carrés, l'estimée de $\mathbf{M}$ optimale est $\widehat{\mathbf{M}}_{LS}$ de (3.10). Nous avons aussi montré que les estimées de $\mathbf{X}$ et $\mathbf{Y}$ associées sont les $\widehat{\mathbf{X}}_{LS}$ et $\widehat{\mathbf{Y}}_{LS}$ de (3.12), et que $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ doivent être de rang plein (condition nécessaire et suffisante) pour que les solutions du problème des moindres carrés totaux (3.11) soient uniques.

3.1.3 Récupération des phases

L'objectif de la présente section est de trouver les facteurs de phase $\{e^{i\widehat{\xi}_\ell}\}_\ell$ tels que, étant donné les vecteurs, $\widehat{\boldsymbol{x}}_\ell, \widehat{\boldsymbol{y}}_\ell \;\; \ell \in \{1, ..., n_i(n_s-1)\}$, il existe (au moins) une matrice unitaire $\mathbf{M}$ qui réalise (3.6).

Nous utilisons le fait que les matrices unitaires préservent le produit scalaire :

$$\begin{aligned} \widehat{\boldsymbol{x}}_{\ell_1}^* \widehat{\boldsymbol{x}}_{\ell_2} &= (\mathbf{M}\widehat{\boldsymbol{x}}_{\ell_1})^* \, (\mathbf{M}\widehat{\boldsymbol{x}}_{\ell_2}) \\ &= \widehat{\boldsymbol{y}}_{\ell_1}^* \widehat{\boldsymbol{y}}_{\ell_2} e^{i(\xi_{\ell_2} - \xi_{\ell_1})}. \end{aligned} \tag{3.13}$$

Par conséquent, pour tout couple $\{\ell_1, \ell_2\}$ dans $\{1, ..., n_i(n_s-1)\}^2$, tel que $\widehat{\boldsymbol{y}}_{\ell_1}^* \widehat{\boldsymbol{y}}_{\ell_2} \neq 0$, nous avons l'estimation suivante de $\xi_{\ell_2} - \xi_{\ell_1}$:

$$\widehat{\xi}_{\ell_1,\ell_2} = arg\left(\frac{\widehat{\boldsymbol{x}}_{\ell_1}^* \widehat{\boldsymbol{x}}_{\ell_2}}{\widehat{\boldsymbol{y}}_{\ell_1}^* \widehat{\boldsymbol{y}}_{\ell_2}} \right) \tag{3.14}$$

où arg est la phase (entre 0 et 2π) d'un nombre complexe. En utilisant ceci, nous pouvons calculer des estimations $\widehat{\xi}_{\ell_2}$ de toutes les phases ξ_{ℓ_2} $\forall \ell_2 \in \{1, ..., n_i(n_s - 1)\}$ relatives à une seule phase ξ_{ℓ_0} (en fixant l_1 à l_0). En utilisant le fait que $\xi_{l_0}^{QST} = 0$, l'estimation de ξ_{ℓ_2} est :

$$\widehat{\xi}_{\ell_2} = \widehat{\xi}_{\ell_0,\ell_2} = arg\left(\frac{\widehat{\boldsymbol{x}}_{\ell_0}^* \widehat{\boldsymbol{x}}_{\ell_2}}{\widehat{\boldsymbol{y}}_{\ell_0}^* \widehat{\boldsymbol{y}}_{\ell_2}}\right). \tag{3.15}$$

Le seul problème qui subsiste est le choix de l_0. Pour le résoudre, examinons (3.14) avec ℓ_1 remplacé par ℓ_0 (c'est ainsi que nous calculons $\widehat{\xi}_{\ell_0,\ell_2}$) :

$$\widehat{\xi}_{\ell_0,\ell_2} = arg\left(\frac{\widehat{\boldsymbol{x}}_{\ell_0}^* \widehat{\boldsymbol{x}}_{\ell_2}}{\widehat{\boldsymbol{y}}_{\ell_0}^* \widehat{\boldsymbol{y}}_{\ell_2}}\right). \tag{3.16}$$

Cela suppose que les produits scalaires $\widehat{\boldsymbol{x}}_{\ell_0}^* \widehat{\boldsymbol{x}}_{\ell_2}$ et $\widehat{\boldsymbol{y}}_{\ell_0}^* \widehat{\boldsymbol{y}}_{\ell_2}$ ne sont pas nuls. En pratique, pour que $\widehat{\xi}_{\ell_1,\ell_2}$ soit une bonne estimation, il faut que les deux produits scalaires soient aussi éloignés de zéro que possible. Il est intéressant de noter que les deux produits scalaires sont censés avoir le même module (voir (3.13)), nous choisissons donc l'indice ℓ_0 solution de :

$$\ell_0 = \arg\max_{\ell_0}\left(\min_{\ell_2} |\widehat{\boldsymbol{y}}_{\ell_0}^* \widehat{\boldsymbol{y}}_{\ell_2}|\right). \tag{3.17}$$

Avec ce choix, ℓ_0 est tel que le plus petit des produits scalaires est le plus grand possible. L'optimisation se fait par recherche exhaustive.

En pratique, si le maximum correspondant est 0, c'est-à-dire si, pour tous les $\widehat{\boldsymbol{y}}_{\ell_0}$, on peut trouver un $\widehat{\boldsymbol{y}}_{\ell_2}$ orthogonal, alors notre méthode pour retrouver les phases ne fonctionne pas car (3.16) ne peut pas s'écrire pour toutes les phases. En pratique, deux vecteurs ne seront jamais vraiment orthogonaux, donc dans notre implémentation de notre méthode, nous considérons que deux vecteurs unitaires sont orthogonaux quand le module de leur produit scalaire est plus petit que b_{orth} initialement fixé à $0,05$. Si le maximum trouvé de (3.17) est plus grand que ce b_{orth}, nous calculons simplement toutes les phases avec (3.16) et ensuite (3.15). Sinon, nous appliquons l'algorithme suivant :

1. Nous commençons par calculer les différences de phase $\widehat{\xi}_{\ell_0,\ell_2}$ (3.16) telles que $\widehat{\boldsymbol{y}}_{\ell_2}$ ne sont pas orthogonales (module du produit scalaire $> b_{orth}$) à $\widehat{\boldsymbol{y}}_{\ell_0}$. Nous obtenons alors une phase absolue pour $\widehat{\boldsymbol{y}}_{\ell_2}$ en utilisant (3.15).

2. Nous appelons $\mathcal{F}$ l'ensemble des $\{\widehat{\boldsymbol{y}}_{\ell_2}\}_{\ell_2}$ pour lesquels nous n'avons pas encore de phase. Nous définissons $\mathcal{S}$ comme le complément de $\mathcal{F}$ dans $\{\widehat{\boldsymbol{y}}_{\ell_2}\}_{\ell_2}$.

3. Pour chaque élément $\widehat{\boldsymbol{y}}_{\ell_f}$ de $\mathcal{F}$

   (a) Si tous les éléments de $\mathcal{S}$ sont orthogonaux (module du produit scalaire $< b_{orth}$) à $\widehat{\boldsymbol{y}}_{\ell_f}$, nous ne changeons rien et passons au $\widehat{\boldsymbol{y}}_{\ell_f}$ suivant.

   (b) Sinon, nous définissons $\widehat{\boldsymbol{y}}_{\ell_s}$ comme l'élément de $\mathcal{S}$ qui est le moins orthogonal (plus grand module du produit scalaire) à $\widehat{\boldsymbol{y}}_{\ell_f}$.

   (c) Nous calculons la phase relative $\widehat{\xi}_{\ell_s,\ell_f}$ avec (3.14) et en déduisons la phase $\widehat{\xi}_{\ell_f}$: $\widehat{\xi}_{\ell_f} = \widehat{\xi}_{\ell_0,\ell_f} = \widehat{\xi}_{\ell_0,\ell_s} - \widehat{\xi}_{\ell_s,\ell_f}$.

   (d) Nous retirons $\widehat{\boldsymbol{y}}_{\ell_f}$ de $\mathcal{F}$ et l'insérons dans $\mathcal{S}$.

4. Si $\mathcal{F}$ est vide, nous avons terminé.

5. Si $\mathcal{F}$ n'est pas vide mais que le nombre d'éléments qu'il contient a diminué depuis l'étape 3, nous allons à l'étape 2.

6. Si $\mathcal{F}$ n'est pas vide et que le nombre d'éléments n'a pas changé, mais que les éléments de $\mathcal{S}$ forment une famille génératrice de $\mathbb{C}^d$, nous retirons de $\widehat{\mathbf{Y}}$ tous les éléments de $\mathcal{F}$, idem pour $\widehat{\mathbf{X}}$ avec les éléments associés. Les nouvelles matrices $\widehat{\mathbf{X}}$ et$\widehat{\mathbf{Y}}$ ont moins de colonnes mais elles sont quand même de rang plein (ce qui garantit que notre méthode de la section 3.1.2 fonctionne), nous quittons donc l'algorithme de récupération de phase et passons à la résolution (3.10) sans les éléments que nous avons supprimés.

7. Si les conditions des étapes 5 et 6 ne sont pas satisfaites, et que b_{orth} vaut $0,05$ (comme au début de l'algorithme), nous changeons b_{orth} en 0 et passons à l'étape 3.

8. Si b_{orth} était déjà égal 0 à l'étape 7, il s'agit d'un cas d'échec et nous sortons de l'algorithme.

Cet algorithme ne peut pas être bloqué dans une boucle infinie :

- Le nombre maximum de fois où l'on peut passer de l'étape 5 à l'étape 3 sans aller à l'étape 6 est le cardinal de $\mathcal{F}$ à l'étape 2 (car le cardinal de $\mathcal{F}$ diminue d'au moins un à chaque fois) qui est strictement plus petit que $n_i(n_s-1)$.
- Le nombre maximum de fois où l'on peut passer de l'étape 7 à l'étape 3 est 1.

L'algorithme peut échouer à l'étape 8, mais nous montrerons dans la section 3.2 qu'il fonctionne toujours si la QPT est possible.

3.1.4 Extension à la QPT standard

La configuration de la SQPT où les états d'entrée sont connus et où un seul délai est considéré est plus standard dans la littérature que la configuration que nous avons introduite. Notre algorithme peut facilement être adapté à cette configuration, car, comme nous l'avons indiqué dans la section précédente, la configuration virtuelle de la figure 3.3 est presque la configuration de la SQPT, avec $n_x = n_i(n_s-1)$ états d'entrée : $\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_x}$. La seule différence entre les deux est que, pour la configuration de la SQPT, les états d'entrée sont considérés comme connus (et non estimés par de QST). Ce n'est pas un problème pour notre algorithme de QPT, nous pouvons toujours définir les matrices $\widehat{X}$ (parce que les états d'entrée sont connus) et $\widehat{Y}$ (parce que les états de sortie sont estimés par l'algorithme de QST), calculer $\widetilde{Y}$ avec l'algorithme de la section 3.1.3 et, enfin, calculer $\widehat{M}_{LS}$ avec (3.10).

3.2 Choix des états initiaux cibles et du nombre d'étapes

3.2.1 Condition nécessaire et suffisante pour l'identifiabilité du système

L'erreur de QST $\{\varepsilon_{j,k}^{QST}\}_{j,k}$ est négligée dans la présente section, donc $\widetilde{\mathbf{Y}} = \mathbf{M}\widehat{\mathbf{X}}$, avec $\widetilde{\mathbf{Y}} = \widehat{\mathbf{Y}}\mathbf{D}(\boldsymbol{\xi})$, et $\mathbf{D}(\boldsymbol{\xi})$ est la matrice diagonale définie par les phases contenues dans $\boldsymbol{\xi}$: $\mathbf{D}(\boldsymbol{\xi}) = \begin{pmatrix} e^{i\xi_1} & & \\ & \ddots & \\ & & e^{i\xi_{n_x}} \end{pmatrix}$ (n_x est le nombre de colonnes de $\widehat{\mathbf{X}}$, donc $n_i(n_s-1)$ dans le problème de base).

Par conséquent, le problème de QPT après de QST consiste à trouver la matrice unitaire $\mathbf{M}$ soumise à l'équation suivante :

$$\widehat{\mathbf{Y}}\mathbf{D}(\boldsymbol{\xi}) = \mathbf{M}\widehat{\mathbf{X}} \tag{3.18}$$

où $\widehat{\mathbf{X}}$ et $\widehat{\mathbf{Y}}$ sont connus grâce à la QST et les éléments $\xi_1, ..., \xi_{n_x}$ de $\boldsymbol{\xi}$ qui définissent $\mathbf{D}(\boldsymbol{\xi})$ sont inconnus avant la QPT.

La condition suivante sur $\mathbf{X}$ est nécessaire et suffisante pour que la QPT soit possible avec la configuration de la figure 3.1 :

$$\forall \ell \in \{1, ..., n_x\},\ \text{rank}\left(\mathbf{F}_S^{n_x}(\boldsymbol{x}_\ell)\right) = d \tag{3.19}$$

où $\mathbf{F}_S$ est la fonction qui prend en entrée une matrice $\mathbf{X}_{in}$ dont les colonnes sont des colonnes de $\mathbf{X}$ et qui retourne les colonnes de $\mathbf{X}$ (groupées dans une matrice dans l'ordre dans lequel elles apparaissent dans $\mathbf{X}$) qui ne sont pas orthogonales à au moins un élément des colonnes de $\mathbf{X}_{in}$. $\mathbf{F}_S^{n_x}$ est $\mathbf{F}_S$ appliqué n_x fois, et $\boldsymbol{x}_\ell$ est la ℓ-ième colonne de $\mathbf{X}$.

Cette définition est étroitement liée à notre algorithme de récupération de phase. Si le b_{orth} de départ a été changé en 0, alors, à l'étape 2, les colonnes de $\mathcal{S}$ correspondent aux colonnes de $\mathbf{F}_S^1(\boldsymbol{x}_{\ell_0})$. Elles "correspondent" dans le sens où il y a autant de colonnes dans $\mathcal{S}$ que dans $\mathbf{F}_S^1(\boldsymbol{x}_{\ell_0})$ et que les positions des colonnes dans $\widehat{\mathbf{Y}}$ et $\mathbf{X}$ respectivement sont les mêmes, parce que $\widehat{\boldsymbol{y}}_j \perp \widehat{\boldsymbol{y}}_j \Leftrightarrow \boldsymbol{x}_j \perp \boldsymbol{x}_j$ ($\perp$ signifie que deux vecteurs sont orthogonaux). La k-ième fois que nous passons à l'étape 5, les colonnes de $\mathcal{S}$ correspondent aux colonnes de $\mathbf{F}_S^{k+1}(\boldsymbol{x}_{\ell_0})$, et, les deux sous espaces linéaires qui contiennent les combinaisons linéaires des colonnes de $\mathcal{S}$ et $\mathbf{F}_S^{k+1}(\boldsymbol{x}_{\ell_0})$ sont les mêmes à une multiplication par $\mathbf{M}$ et un re-phasage près, ils ont donc la même dimension (c'est important).

L'équation (3.19) est une condition sur les colonnes de la matrice $\mathbf{X}$. Cependant, il est très facile de vérifier que nous aurions pu définir cette condition avec les colonnes de $\widehat{\mathbf{X}}$, $\mathbf{Y}$ ou $\widehat{\mathbf{Y}}$ et obtenir une condition équivalente. En effet, multiplier toutes les colonnes par la même matrice unitaire à gauche, ou multiplier chacune d'entre elles par un facteur de phase scalaire différent ne change ni le rang ni l'orthogonalité entre les colonnes. Nous utilisons $\mathbf{X}$ car il s'agit d'une matrice qui contient les états d'entrée (du circuit virtuel de la figure 3.3), et nous préférons avoir une condition sur les états d'entrée.

L'annexe B.3 montre que, en l'absence d'erreur de QST, (3.19) est une condition nécessaire (dans B.3.2) et suffisante (dans B.3.1) pour que la QPT soit possible (c'est-à-dire pour que $\mathbf{M}$ soit identifiable à une phase globale près). De plus, la preuve de la suffisance de (3.19) dans B.3.1 montre également que notre algorithme de QPT fonctionne toujours si (3.19) est vraie.

Il s'agit d'une validation forte de notre algorithme, qui réussit toujours à réaliser la QPT si la configuration (c'est-à-dire les états mesurés) la rend possible et n'échoue que si la QPT était impossible avec les états mesurés (c-à-d si quel que soit l'algorithme que l'on pourrait imaginer, la QPT ne pourrait pas être réalisée car le problème n'est pas injectif, il existe plusieurs processus distincts compatibles avec les mesures). Bien entendu, cela n'est vrai que s'il n'y a pas d'erreur de QST. Lorsque l'on considère des erreurs de QST non nulles, on peut avoir des problèmes avec des configurations pour lesquelles $\widehat{\mathbf{X}}$ est mal conditionnée et pour lesquelles $\widehat{\mathbf{X}}$ contient des groupes de colonnes qui sont trop proches d'être orthogonales pour que la récupération de la phase réussisse.

3.2.2 Une condition nécessaire plus simple

Nous avons établi que (3.19) est une condition nécessaire et suffisante pour que la QPT soit possible.

La condition (3.19) est cependant assez lourde à vérifier, et nous préférons utiliser la condition suffisante suivante (on peut montrer qu'elle n'est pas nécessaire) :

$$\begin{gathered}\text{rang}(\mathbf{X}) = d \quad \text{et} \\ \exists \ell_0 \in \{1, ..., n_x\},\ \text{t.q. } \forall \ell \in \{1, ..., n_x\}\ \boldsymbol{x}_{\ell_0} \not\perp \boldsymbol{x}_\ell.\end{gathered} \tag{3.20}$$

En clair, $\mathbf{X}$ est de rang plein et il existe une colonne de $\mathbf{X}$ qui n'est orthogonale à aucune des autres.

Il est très facile de vérifier que (3.20)⇒ (3.19) : si (3.20) est vrai, alors toutes les colonnes de $\mathbf{X}$ sont dans $\mathbf{F}_S^k(\boldsymbol{x}_\ell)$ pour tout $k \geq 2$ et tout ℓ.

Par conséquent, lorsque nous concevons la configuration de QPT de la figure 3.1, nous devons espérer (ou nous assurer) que les états quantiques représentés par les colonnes de $\mathbf{X}$ satisfont (3.19) ou (3.20).

En pratique, (3.20) est une condition très raisonnable. La probabilité que deux états aléatoires (avec n'importe quelle fonction de densité non dégénérée) soient orthogonaux est de 0 et la probabilité que d (ou plus) vecteurs aléatoires (dans un espace de Hilbert à d dimensions) soient dans un sous-espace de dimension $\leq d-1$ est également de 0. Donc les états qui font échouer notre algorithme (qui sont "trop orthogonaux" ou ne sont pas de rang plein) sont dans un ensemble de mesure nulle.

Par conséquent, les conditions (3.20) (et donc la condition (3.19)) seront toujours satisfaites en pratique. Même si nous essayons de préparer des états qui font échouer notre méthode de QPT, la petite erreur aléatoire dans leur préparation garantira que les états réels satisfont (3.20). Cela ne signifie pas pour autant que nous pouvons ignorer les conditions sans problème. En effet, si $\mathbf{X}$ est de rang plein mais est presque singulière, ou s'il y a trop de colonnes proches d'être orthogonales, alors notre algorithme devrait mal fonctionner.

3.2.3 Nos recommendations pour le choix des états initaux

Si possible, nous recommandons de considérer ce que la porte que nous essayons d'identifier est supposée faire (sinon nous proposons ci-après une solution qui fonctionne pour n'importe quelle porte unitaire sans connaissance préalable). Par exemple, si la porte à identifier est supposée être une porte CNOT à 2 qubits, $\mathbf{M}_{tg} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ ("tg" signifie cible), alors, choisir un seul état d'entrée ($n_i = 1$) et $n_s = d+1 = 5$ pas de temps est une très mauvaise idée. En effet, la porte CNOT appliquée deux fois est censée retourner l'état initial. Par conséquent, pour $n_s = 5$, la matrice $\mathbf{X}$ possède deux paires de colonnes identiques. En pratique, $\mathbf{X}$ sera presque singulier et la qualité de l'estimation de $\mathbf{X}$ sera très mauvaise.

Pour avoir une matrice $\mathbf{X}$ telle que (3.20) soit "confortablement" satisfaite par la porte CNOT, considérons $n_s = 2$ délais, et $n_i = d = 4$ états d'entrée qui forment une base de l'espace de Hilbert avec l'un d'entre eux loin d'être orthogonal à tous les autres. Par exemple, nous pouvons viser ces cibles :

$$\boldsymbol{v}_1^{tg} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{v}_2^{tg} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{v}_3^{tg} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \boldsymbol{v}_4^{tg} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{3.21}$$

Les états réels $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4$ devraient être raisonnablement proches de ces cibles, mais notre algorithme de QST se comportera comme si les états étaient totalement inconnus (de sorte que nous soyons résistants aux erreurs systématiques).

En utilisant le fait que la multiplication de deux vecteurs par la même matrice unitaire préserve leur produit scalaire, il est très facile de vérifier que, $\mathbf{X} = \mathbf{M}\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix}$ satisfait (3.20) et donc (3.19) avec une marge confortable pour toute matrice unitaire $\mathbf{M}$ si les $\{\boldsymbol{v}_k\}_k$ sont égaux (ou assez proches) de leur cible.

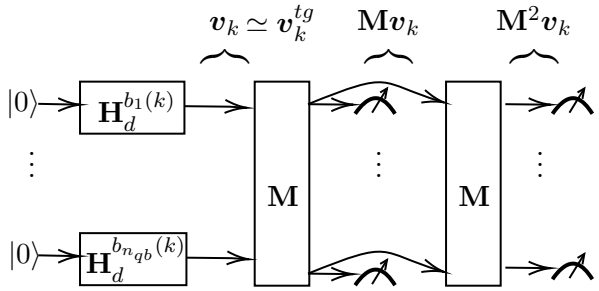


FIGURE 3.4 : Circuit quantique représentant la préparation de l'état $\boldsymbol{v}_k^{tg}$ de (3.22) et la QPT semi-aveugle de $\mathbf{M}$ utilisant cet état. Le circuit doit être réalisé pour tous les $k \in \{1, ..., d\}$. La matrice $\mathbf{H}_d$ est la matrice unitaire associée à la porte de Hadamard à un qubit, elle est de taille 2. Et $b_j(k) \in \{0, 1\}$ est le j-ième élément de la décomposition binaire de $k-1$ sur n_{qb} bits. Ainsi, $\mathbf{H}_d^{b_j(k)} = \mathbf{I}_2$ si $b_j(k) = 0$ et $\mathbf{H}_d^{b_j(k)} = \mathbf{H}_d$ si $b_j(k) = 1$. Cette configuration permet de réaliser la QPT pour n'importe quelle valeur de la matrice unitaire $\mathbf{M}$ qui représente la porte à identifier. Notre algorithme est robuste à une mauvaise implémentation des portes de Hadamard. Les "doubles flèches" au milieu symbolisent le fait que la moitié des copies est mesurée (flèche droite) et que l'autre moitié est réintroduite dans la porte (flèche courbée).

Les états cibles de (3.21) peuvent être généralisés à n'importe quel nombre de qubits n_{qb} :

$$[\boldsymbol{v}_1^{tg}, ..., \boldsymbol{v}_d^{tg}] = \underbrace{\begin{pmatrix} 1 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \otimes ... \otimes \begin{pmatrix} 1 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix}}_{n_{qb} \text{ fois}} \tag{3.22}$$

dans le sens où les vecteurs cibles du côté gauche de l'égalité sont définis comme les colonnes de la matrice du côté droit. Pour toutes les portes unitaires à n_{qb} qubits (i.e. $\forall \mathbf{M}$), l'utilisation des états de (3.22) avec $n_i = d, n_s = 2$ génère une matrice $\mathbf{X}$ qui satisfait (3.20) parce que ses colonnes forment une base et qu'aucune d'entre elles n'est orthogonale à une autre. Ces états ont l'avantage de ne pas être intriqués et d'être très faciles à préparer : si tous les qubits sont initialisés à $|0\rangle$, alors le k-ième état peut être préparé en appliquant une porte de Hadamard à 1 qubit aux qubits dont les indices sont tels qu'il y a un 1 à ces indices dans la décomposition binaire de $k-1$. Par exemple, pour $\boldsymbol{v}_1^{tg}$, tous les qubits restent égaux $|0\rangle$ et aucune porte de Hadamard n'est appliquée, pour $\boldsymbol{v}_2^{tg}$, tous les qubits sont initialisés à $|0\rangle$ et une porte de Hadamard est appliquée au dernier qubit, etc.

La configuration de la figure 3.4 avec $n_i = d$, $n_s = 2$ génère les états initiaux de (3.22), elle est très intéressante :

- Comme expliqué ci-dessus, elle peut identifier n'importe quel type de porte unitaire sans connaissance préalable (la condition suffisante (3.20) est toujours satisfaite). Nous avons introduit cette configuration dans le cas où on sait que le processus à identifier est censé être une porte CNOT (car c'est un cas compliqué), mais elle est bien plus générale.
- Elle est assez facile à préparer, car nous n'avons besoin que d'un seul type de porte (Hadamard) autre que la porte que nous voulons identifier.
- Nous pouvons également tolérer des erreurs dans les portes de Hadamard : il n'est pas nécessaire qu'elles soient parfaites ni qu'elles soient toutes identiques ; nous exigeons simplement que le comportement de chaque porte reste le même pendant que nous préparons des copies de chaque état. Il faut aussi que les états crées par les portes continuent de vérifier (3.19), mais nous verrons, grâce aux tests de la section 5.1.2, que c'est généralement le cas en pratique.

- L'utilisation de ces portes (simples) au plus une fois pour chaque qubit devrait limiter les problèmes de décohérence.

Un inconvénient est qu'il peut y avoir des problèmes avec le conditionnement de $\mathbf{X}$ quand n_{qb} augmente. La matrice $\mathbf{X}$ est toujours inversible avec les états de (3.22). Cependant, nous avons observé que, en pratique, le conditionnement de $\mathbf{X}$ décroit de façon exponentielle avec le nombre de qubits. Pour plus de 4 qubits, nous recommandons de considérer plus de d états d'entrée, ou d'utiliser des portes de rotation à 1 qubit (au lieu de portes de Hadamard) avec des angles adaptés qui fournissent un conditionnement raisonnable et maintiennent la condition de non-orthogonalité dans (3.20). Le conditionnement et le produit scalaire des colonnes de $\mathbf{X}$ ne dépendent pas de la valeur de la matrice unitaire $\mathbf{M}$, ils ne dépendent que de la valeur des états initiaux.

Cette configuration (figure 3.4) peut être moins efficace que celle décrite ci-dessous (figure 3.5) si nous avons une idée approximative de ce que la porte que nous essayons d'identifier est censée faire. Par exemple, avec $n_{qb} = 2$, si le processus unitaire que nous voulons identifier est censé être représenté par

$$\mathbf{M}_{tg} = \frac{1}{2}\begin{pmatrix} 1 & -\sqrt{2} & 0 & 1 \\ 1 & \sqrt{2} & 0 & 1 \\ 1 & 0 & -\sqrt{2} & -1 \\ 1 & 0 & \sqrt{2} & -1 \end{pmatrix}, \tag{3.23}$$

montrons comment exploiter cette connaissance pour concevoir une configuration plus efficace. Si $\mathbf{M}$ est suffisamment proche de $\mathbf{M}_{tg}$, alors nous pouvons utiliser un seul état initial, $n_i = 1$ et $n_s = 5$ délais, et $\mathbf{M}$ devrait (sauf s'il est vraiment loin de la cible) faire évoluer l'état initial de manière à ce que (3.20) soit vraie. Si l'état initial est $\boldsymbol{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ par exemple, nous pouvons calculer la matrice $\mathbf{X}$ que nous aurions si $\mathbf{M}$ était exactement la cible :

$$\mathbf{X}_{tg} = \begin{bmatrix} \mathbf{M}_{tg}\boldsymbol{v}_1 & \dots & \mathbf{M}_{tg}^d\boldsymbol{v}_1 \end{bmatrix}. \tag{3.24}$$

La matrice $\mathbf{X}_{tg}$ est bien conditionnée (la plus grande valeur singulière n'est que 2.5 fois plus grande que la plus petite) et sa première colonne est raisonnablement loin d'être orthogonale aux autres (plus petit module du produit scalaire de $0,14$). Cette configuration (généralisée pour un nombre quelconque de qubits) est représentée à la figure 3.5.

Le fait de considérer moins d'états et plus de délais que dans la configuration de la figure 3.4 signifie que nous nous appuyons en partie sur la porte que nous essayons d'identifier pour créer les états que nous utiliserons. Les inconvénients de cette méthode sont les suivants :

- Nous ne sommes jamais sûrs que (3.19) est vraie avec une marge confortable, à moins que nous ayons une idée à peu près correcte de ce que fait la porte que nous voulons caractériser. En revanche, avec $n_i = d, n_s = 2$ et les états initiaux de (3.22), nous sommes sûrs que (3.20) et (3.19) sont confortablement satisfaits pour tout $\mathbf{M}$.

- Elle ne fonctionne pas avec la plupart des portes quantiques "classiques" (c'est-à-dire les plus souvent considérées et utilisées dans la littérature) parce que ces portes impliquent souvent des rotations de 90° (ce qui peut rendre trop de colonnes de $\mathbf{X}$ presque orthogonales) ou ne changent pas certaines directions de l'espace de Hilbert (ce qui peut rendre $\mathbf{X}$ mal conditionnée).

- Des valeurs plus élevées de n_s peuvent créer des problèmes de décohérence pour certaines architectures, car l'état est observé après avoir attendu $n_s\Delta_t$.

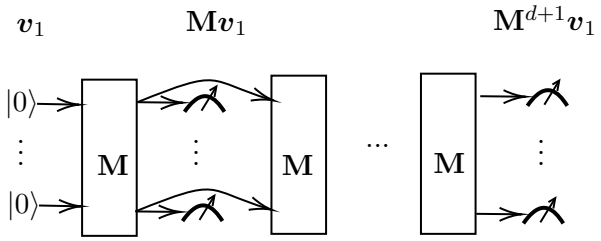


FIGURE 3.5 : Circuit quantique qui peut être utilisé pour effectuer la QPT de la porte représentée par $\mathbf{M}$ avec un seul état d'entrée et $d+1$ pas de temps : $n_i = 1, n_s = d+1$. Cette configuration fonctionne si et seulement si la matrice $\mathbf{M}$ génère des états $\mathbf{M}\boldsymbol{v}_1, ..., \mathbf{M}^d\boldsymbol{v}_1$ qui satisfont (3.19). Cette configuration est très facile à réaliser car il n'y a qu'une seule valeur de l'état initial. Les "doubles flèches" symbolisent que $n_c n_t$ copies sont mesurées (flèche droite) et que les autres sont réintroduites dans la porte (flèche courbée).

Cependant, la configuration de la figure 3.5 présente les avantages suivants :

- La préparation de copies d'une seule valeur d'état d'entrée est beaucoup plus simple pour l'opérateur. Un seul type de porte (la porte à identifier) est utilisé. L'utilisation de plusieurs types de portes est problématique, car même si l'algorithme de QPT de la section 3.1 ne fait aucune hypothèse sur les valeurs des matrices unitaires qui représentent chaque porte ($\mathbf{M}$ et les portes utilisées pour la préparation des états initiaux), il suppose quand même qu'il s'agit de portes unitaires qui font exactement la même chose à chaque fois que nous répétons l'expérience. Si ce n'est pas le cas, la qualité de notre estimation de $\mathbf{M}$ sera dégradée.

- Un n_s plus élevé signifie qu'un plus grand nombre d'états de sortie estimés sont "réutilisés" comme estimations des états d'entrée et vice versa. Cela signifie que, en tout, moins d'états sont mesurés et que nous pouvons nous permettre de faire plus de copies de chaque état que nous mesurons. Par exemple, avec $n_{qb} = 2$, nous pouvons utiliser $n_i = 4$ valeurs différentes des états initiaux de (3.22) et $n_s = 2$ pas de temps pour estimer une porte à 2 qubits, ce qui nécessite 8 états différents à mesurer. Si la même porte peut être estimée avec une seule valeur d'état d'entrée ($n_i = 1$) et $n_s = 5$ délais, cela ne nécessite que 5 états différents à mesurer.

- L'exemple de la porte cible (3.23) que nous avons utilisée pour illustrer un cas où $n_i = 1, n_s = d+1$ peut sembler artificiel et faire penser au lecteur que $n_i = 1, n_s = d+1$ ne fonctionne que très rarement, mais il fonctionne assez bien pour les portes unitaires aléatoires (voir la section 5.1.8).

Nous pourrions envisager une configuration intermédiaire avec $d > n_i > 1$. Cela peut être utile si $\mathbf{M}$ est l'identité (à une phase globale près) sur un sous-espace de l'espace de Hilbert mais apporte suffisamment de diversité au supplément de ce sous-espace. Par exemple, supposons que nous voulons effectuer la QPT pour une porte qui devrait être représentée par $\mathbf{M}_{tg} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$. Si nous considérons une seule valeur arbitraire d'état d'entrée $\boldsymbol{v}_1$ et si $\mathbf{M} = \mathbf{M}_{tg}$, alors, le $\mathbf{X}_{tg}$ de (3.24) ne sera jamais de rang plein parce que ses deux premières lignes contiendront la même valeur dans toutes les colonnes, elles seront donc colinéaires. Il est

très facile de vérifier que si nous posons $n_i = 2, n_s = 3$ avec $\boldsymbol{v}_1^{tg} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}, \boldsymbol{v}_2^{tg} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\1\end{pmatrix}$ et si les états d'entrée sont égaux à leur cible, alors, (3.19) est confortablement satisfaite. Et donc (3.19) devrait toujours être satisfaite s'ils sont proches de leur cible.

3.2.4 Lien avec la littérature sur la QPT unitaire

Dans l'annexe 1 de [RGK13], Reich et al. donnent une condition nécessaire et suffisante sur les états d'entrée pour qu'un processus unitaire soit déterminé de manière unique (à une phase globale près) parmi tous les processus (unitaires ou non), cette condition est la suivante

$$Com\left(\{\boldsymbol{\rho}_\ell\}_\ell\right) = \{e^{i\theta}\mathbf{I}_d\}_{\theta\in\mathbb{R}}. \tag{3.25}$$

où $\boldsymbol{\rho}_\ell$ est la matrice densité du ℓ-ièmes état d'entrée (mélange ou pur), Com est le commutant, $Com\left(\{\boldsymbol{\rho}_\ell\}_\ell\right)$ désigne l'ensemble des matrices *unitaires* qui commutent avec tous les $\{\boldsymbol{x}_\ell\boldsymbol{x}_\ell^*\}_\ell$.

Si nous écrivons (3.25) pour les états d'entrée purs : $\boldsymbol{\rho}_\ell = \boldsymbol{x}_\ell\boldsymbol{x}_\ell^*$, la condition devient :

$$Com\left(\{\boldsymbol{x}_\ell\boldsymbol{x}_\ell^*\}_\ell\right) = \{e^{i\theta}\mathbf{I}_d\}_{\theta\in\mathbb{R}} \tag{3.26}$$

Les conditions (3.19) et (3.26) sont équivalentes, nous le montrons dans l'annexe B.3.3. Cela ne signifie pas que (3.19) est équivalent à la condition de de Reich et al. (3.25) car cette dernière a été définie avec des états mélanges en entrée, et (3.26) est sa reformulation avec des états d'entrée purs. Par conséquent, l'équivalence entre (3.19) et (3.25) n'existe que lorsque les états d'entrée sont purs.

Dans [RGK13], les auteurs donnent deux exemples d'ensembles d'états d'entrée qui satisfont (3.25). Le premier est un ensemble de $d+1$ états d'entrée purs : d états qui forment une base orthonormée et un dernier état qui est la moyenne des premiers d (cela fonctionne avec n'importe quelle base orthonormée). On retrouve la même idée que dans (3.20) : l'un des états n'est pas orthogonal aux autres.

Reich et al. montrent également que si l'on autorise l'utilisation d'états mélanges, seuls deux états d'entrée sont nécessaires pour que la QPT soit possible. En effet, si un état mélange avec d valeurs propres distinctes passe par le processus unitaire, la sortie a les mêmes valeurs propres, mais les vecteurs propres sont multipliés par la matrice unitaire associée au processus. Il est donc possible d'évaluer la matrice unitaire sur la base orthonormée des vecteurs propres en utilisant un seul état d'entrée mixte, mais ce n'est pas suffisant car tous les vecteurs propres sont orthogonaux les uns par rapport aux autres (théorème spectral), et il faut donc un deuxième état d'entrée. Nous avons choisi de ne pas considérer les états mélanges parce qu'ils peuvent être difficiles à produire. Baldwin et al. ont noté dans [BKD14], lorsqu'ils utilisent les résultats de [RGK13], que "in practice we do not have reliable procedures to produce a desired, reproducible, mixed state". Nous ne sommes pas vraiment d'accord avec cette affirmation, tout état mélange peut être considéré comme un mélange statistique d'au plus d états purs (représentés par les vecteurs propres de la matrice de densité). Donc, si nous pouvons générer tous les états purs du mélange et ensuite choisir au hasard quel état pur est généré pour chaque copie de l'état mélange que l'on veut créer (avec la probabilité de choisir chaque vecteur propre donnée par la valeur propre associée de la matrice densité de l'état mélange cible), alors, nous pouvons générer un état mélange spécifique. Générer un état mélange de cette manière ne nous servirait cependant à rien, car cela implique de préparer des copies de d états purs différents, puis de les "mélanger" (sans enregistrer quel état est utilisé pour quelle copie de l'état mélange). L'utilisation directe de d états purs est plus efficace que l'utilisation de ce "mélange".

Au-delà de la reformulation de (3.26) en (3.19) et de la définition d'un ensemble d'états d'entrée simples, notre contribution est que, contrairement à [RGK13], nous fournissons un algorithme de QPT qui fonctionne avec tout ensemble d'états d'entrée qui vérifie notre condition nécessaire et suffisante. Nous avons également profité du fait que (3.19) n'est pas très contraignante pour présenter notre configuration semi-aveugle qui élimine le problème des erreurs systématiques lors de la préparation des états d'entrée du processus à identifier.

3.3 Estimation des paramètres du processus unitaire par maximum de vraisemblance

3.3.1 Principe

Dans la section 3.1, nous avons décrit une méthode de QPT basée sur la QST de tous les états mesurés. Dans la présente section, nous décrivons une méthode de QPT directe, qui estime le processus directement à partir des mesures. En pratique, nous cherchons la matrice unitaire $\widehat{\mathbf{M}}_{ML}$ et les états initiaux $\widehat{\mathbf{V}}_{ML} = [\widehat{\mathbf{v}_1}, ..., \widehat{\mathbf{v}_{n_i}}]$ (ces derniers ne nous intéressent pas vraiment mais ils doivent être considérés) qui maximisent la vraisemblance des mesures $\mathbf{N} = \begin{bmatrix} \mathbf{n}_{1,1} & \mathbf{n}_{1,2} & ... & \mathbf{n}_{n_i,n_s} \end{bmatrix}$ où $\mathbf{n}_{j,k}$ contient les $n_x = dn_t$ mesures faites sur l'état $\boldsymbol{v}_{j,k} = \mathbf{M}^k \boldsymbol{v}_j$ qui est le j-ième état initial à l'étape k. Formellement :

$$(\widehat{\mathbf{M}}_{ML}, \widehat{\mathbf{V}}_{ML}) = \underset{\mathbf{M},\mathbf{V}}{\arg\min} \mathcal{L}_{\mathbf{M},\mathbf{V}}(\mathbf{N}) \tag{3.27}$$

où $\mathcal{L}$ est l'opposé de la log-vraisemblance qui est minimisé pour maximiser la vraisemblance. Pour réaliser cette optimisation, nous avons besoin d'un bon point initial (critère non-convexe), pour cela, nous utilisons le $\widehat{\mathbf{M}}_{LS}$ de (3.10). Les colonnes de $\mathbf{V}$ doivent aussi être estimées, pour ce faire, nous allons utiliser le $\mathbf{X}_{LS}$ de (3.12) (qui prend en compte tous les résultats de la QST) et $\widehat{\mathbf{M}}_{LS}$. Nous notons cette estimée $\widehat{\mathbf{V}}_{LS}$, la j-ième colonne de $\widehat{\mathbf{V}}_{LS}$ vaut :

$$\widehat{\boldsymbol{v}}_{LS_j} = \frac{1}{n_s - 1} \sum_{k=1}^{n_s - 1} \widehat{\mathbf{M}}_{LS}^{-k} \widehat{\boldsymbol{x}}_{k+(n_s-1)(j-1)}. \tag{3.28}$$

En pratique, si on utilise cette formule, il y a un problème de phase, les $\widehat{\boldsymbol{x}}$ ne sont bons qu'à des phases globales près, donc quand on les somme (après les avoir multipliés par des puissances de $\widehat{\mathbf{M}}_{LS}$), on somme des quantités qui ne sont exactes qu'à une phase près, les phases vont se sommer destructivement. Pour régler ce problème de phase, on pourrait utiliser les colonnes de $\widetilde{\mathbf{Y}}$ calculé dans la section 3.1.3 qui calculait les phases qui sont cohérentes avec l'unitarité de $\mathbf{M}$, mais en pratique, comme on a une estimée de $\mathbf{M}$, on peut l'utiliser. Pour tout vecteur complexe $\boldsymbol{x}$ de taille d, on définit $\xi_{\boldsymbol{x}}$ comme le facteur de phase suivant $\xi_{\boldsymbol{x}} = e^{\frac{-i}{2} arg(\boldsymbol{x}^t \boldsymbol{x})}$, $\xi_{\boldsymbol{x}}$ est tel que $(\xi_{\boldsymbol{x}}\boldsymbol{x})^T(\xi_{\boldsymbol{x}}\boldsymbol{x})$ est un réel positif (trivial à vérifier). En quelque sorte, $\xi_{\boldsymbol{x}}\boldsymbol{x}$ est une version re-phasée de ξ telle que la somme des carrés de tous les éléments est un réel positif. Il n'existe que deux tels facteurs de phase $\xi_{\boldsymbol{x}}$ et $-\xi_{\boldsymbol{x}}$. Définissons maintenant $\boldsymbol{f}^{reph}_{\widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)}}$ comme la fonction qui change les phases des éléments de la somme de (3.28) pour qu'ils se somment constructivement :

$$\begin{aligned} \boldsymbol{f}^{reph}_{\widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)}} &: \boldsymbol{x} \to (-1)^b \xi_{\boldsymbol{x}} \boldsymbol{x} \text{ où } b \text{ est un booléen (0 ou 1) défini par :} \\ & b = Re\big((\xi_{\widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)}} \widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)})^* (\xi_{\boldsymbol{x}} \boldsymbol{x})\big) > 0 \end{aligned} \tag{3.29}$$

Avec cette définition, la fonction $\boldsymbol{f}^{reph}_{\widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)}}$ peut être utilisée pour que les éléments de la somme de (3.28) se somment de façon non-destructive, (3.28) devient :

$$\widehat{\boldsymbol{v}}_{LS_j} = \frac{1}{n_s - 1} \sum_{k=1}^{n_s-1} \boldsymbol{f}^{reph}_{\widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)}} \left(\widehat{\mathbf{M}}^{-k} \widehat{\boldsymbol{x}}_{k+(n_s-1)(j-1)} \right). \tag{3.30}$$

$\widehat{\mathbf{M}}_{LS}$ a été remplacé par $\widehat{\mathbf{M}}$ pour avoir une expression de l'estimée des états initiaux qui peut être calculée avec n'importe quelle estimée de la matrice $\mathbf{M}$, et, en particulier, pour $\widehat{\mathbf{M}} = \widehat{\mathbf{M}}_{LS}$. Nous aurions pu faire d'autres choix pour la fonction de re-phasage $\boldsymbol{f}^{reph}_{\widehat{\boldsymbol{x}}_{1+(n_s-1)(j-1)}}$ (il existe d'autres fonctions de re-phasage qui font que deux vecteurs se somment constructivement), mais le choix que nous avons fait est très résistant aux erreurs de QST.

Dans les sections suivantes, nous verrons comment la vraisemblance est calculée (section 3.3.2) et comment elle est minimisée (section 3.3.3)

3.3.2 Calcul de la vraisemblance des mesures pour un processus et des états initiaux donnés

On ré-écrit l'expression de la vraisemblance du vecteur des résultats de mesures $\boldsymbol{n}_{j,k}$, sur un seul état $\boldsymbol{v}_{j,k}$ ($\boldsymbol{v}_{j,k} = \mathbf{M}^k \boldsymbol{v}_j$ est le j-ième état initial après $k\Delta_t$) du modèle multinomial de (2.22), en remplaçant $\boldsymbol{v}(\boldsymbol{x}', \boldsymbol{y}')$ par $\boldsymbol{v}_{j,k}$ (car nous n'utilisons pas la même paramétrisation ici que pour la QST) :

$$\mathcal{L}^{exact}_{\boldsymbol{v}_{j,k}}(\boldsymbol{n}_{j,k}) = -\sum_{\ell=1}^{n_{\mathbf{A}}} (\boldsymbol{n}_{j,k})_\ell log((|\mathbf{A}\boldsymbol{v}_{j,k}|^2)_\ell) \tag{3.31}$$

où $n_{\mathbf{A}} = dn_t$ est le nombre d'éléments de $\boldsymbol{n}_{j,k}$ (ou le nombre de lignes de $\mathbf{A}$), et $(.)_\ell$ est le ℓ-ième élément d'un vecteur.

Nous avons aussi défini le modèle gaussien (régularisé) dans (2.24). Avec ce modèle la vraisemblance des résultats de mesures $\boldsymbol{n}_{j,k}$ pour l'état $\boldsymbol{v}_{j,k}$ s'écrit avec le vecteur d'erreur de mesures $\varepsilon_{j,k} = \frac{\boldsymbol{n}_{j,k}}{n_c} - |\mathbf{A}\boldsymbol{v}_{j,k}|^2$, on peut la ré-ecrire sans $\varepsilon_{j,k}$:

$$\mathcal{L}^{gauss}_{\boldsymbol{v}_{j,k}}(\boldsymbol{n}_{j,k}) = \sum_{\ell=1}^{n_{\mathbf{A}}} \frac{\left((\boldsymbol{n}_{j,k})_\ell - n_c(|\mathbf{A}\boldsymbol{v}_{j,k}|^2)_\ell\right)^2}{n_c(\widetilde{\boldsymbol{p}_{j,k}})_\ell}. \tag{3.32}$$

où $\widetilde{\boldsymbol{p}_{j,k}} = \frac{\boldsymbol{n}+5}{n_c+5d}$ contient les probabilités régularisées.

Ces vraisemblances concernent les mesures sur un seul état. Nous voulons calculer la vraisemblance des mesures sur tous les états pour résoudre le problème (3.27). Définissons l'ensemble des nombres d'occurences de tous les résultats de mesure observés $\mathbf{N} = \begin{bmatrix} \mathbf{n}_{1,1} & \mathbf{n}_{1,2} & \dots & \mathbf{n}_{n_i,n_s} \end{bmatrix}$, et calculons sa vraisemblance. Notons que, pour des paramètres $\mathbf{M}$ et $\mathbf{V}$ donnés (qui déterminent les paramètres des lois des mesures), les mesures effectuées sur différents états sont indépendantes, on peut donc sommer les opposés des log-vraisemblances de (3.31) ou (3.32) :

$$\mathcal{L}^s_{\mathbf{V},\mathbf{M}}(\mathbf{N}) = \sum_{j=1}^{n_i} \sum_{k=1}^{n_s} \mathcal{L}^s_{\mathbf{M}^k \boldsymbol{v}_j}(\mathbf{n}_{j,k}) \tag{3.33}$$

où $s \in \{gauss, exact\}$ dépend du modèle de vraisemblance que l'on utilise.

En pratique, on veut calculer $\min_{\mathbf{M},\mathbf{V}} \mathcal{L}^s_{\mathbf{M},\mathbf{V}}(\mathbf{N})$, ce problème est équivalent à $\min_{\mathbf{M}} \min_{\boldsymbol{v}_1,\dots,\boldsymbol{v}_{n_i}} \mathcal{L}^s_{\mathbf{M},\mathbf{V}}(\mathbf{N})$, et, avec (3.33), ce problème se simplifie :

$$\widehat{\mathbf{M}}_{ML} = \arg\min_{\mathbf{M}} \sum_{j=1}^{n_i} \min_{\boldsymbol{v}_k} \sum_{k=1}^{n_s} \mathcal{L}^s_{\mathbf{M}^k \boldsymbol{v}_j}(\mathbf{n}_{j,k}) \tag{3.34}$$

En résumé, la vraisemblance de l'ensemble des mesures est définie avec l'équation (3.33) où $\mathcal{L}^s_{\mathbf{M}^k \boldsymbol{v}_j}(\mathbf{n}_{j,k})$ est définie par (3.31) ou (3.32) en fonction du modèle que l'on choisit pour

la vraisemblance. Pour minimiser $\mathcal{L}$, on choisit de séparer les deux paramètres selon lesquels on minimise ($\mathbf{M}$ et $\mathbf{V}$) avec (3.34). Cela permet de simplifier la minimisation sur $\mathbf{V}$ en n_i minimisations sur $\{\boldsymbol{v}_k\}_{k\in\{1,...,n_i\}}$, ce qui est très avantageux d'un point de vue calculatoire : résoudre n_s problèmes d'optimisations à $2n_{qb}$ paramètres réels (un vecteur d'état $\boldsymbol{v}_k$ de taille d non-intriqué a $2n_{qb}$ paramètres, nous allons l'expliquer dans la section 3.3.3), est beaucoup plus simple que de résoudre un seul problème d'optimisation à $2n_{qb}n_i$ paramètres réels ($\mathbf{V}$ contient les n_i vecteurs d'état initiaux). Le problème que l'on va résoudre avec un algorithme de type descente de gradient est donc (3.34). Nous allons à présent expliquer comment les arguments $\mathbf{M}$ et $\boldsymbol{v}_j$ sont paramétrés avec le nombre optimal de paramètres réels et sans contraintes.

3.3.3 Paramétrisation des arguments

Pour un $j \in \{1, ..., d\}$ donné, $\mathbf{v}_j$ représente un état non-intriqué. Par définition de l'intrication, cet état peut être décomposé en n_{qb} états de 1 qubit : $\boldsymbol{v}_j = \mathbf{q}(r_{j,1}, \theta_{j,1}) \otimes ... \otimes \mathbf{q}(r_{j,n_{qb}}, \theta_{j,n_{qb}})$. Chaque qubit de la décomposition s'écrit avec deux paramètres réels $r_{j,h}$ et $\theta_{j,h}$:

$$\mathbf{q}(r_{j,h}, \theta_{j,h}) = \begin{bmatrix} r_{j,h} \\ \sqrt{1 - r_{j,h}^2} e^{i\theta_{j,h}} \end{bmatrix}. \tag{3.35}$$

Donc, les états $\boldsymbol{v}_j$ peuvent être paramétrés avec $2n_{qb}$ paramètres réels : $\boldsymbol{v}_j = g_{\boldsymbol{v}}(r_{j,1}, \theta_{j,1}, ..., r_{j,n_{qb}}, \theta_{j,n_{qb}})$ avec :

$$\boldsymbol{g_v}(r_{j,1}, \theta_{j,1}, ..., r_{j,n_{qb}}, \theta_{j,n_{qb}}) = \mathbf{q}(r_{j,1}, \theta_{j,1}) \otimes ... \otimes \mathbf{q}(r_{j,n_{qb}}, \theta_{j,n_{qb}}). \tag{3.36}$$

Le problème de $\boldsymbol{g_v}$ est que les n_{qb} premiers arguments sont contraints (ils doivent être entre 0 et 1), nous voulons faire de l'optimisation sans contrainte (plus rapide d'un point de vue temps de calcul). Nous utilisons donc la fonction $\boldsymbol{f_v}$:

$$\boldsymbol{f_v}(h_1, ..., h_{2n_{qb}}) = \boldsymbol{g_v}\left(\frac{1}{2} + \frac{atan(h_1)}{\pi}, h_{n_{qb}+1}, ..., \frac{1}{2} + \frac{atan(h_{n_{qb}})}{\pi}, h_{2n_{qb}}\right) \tag{3.37}$$

où les $h_1, ..., h_{n_{qb}}$ correspondent aux $r_{j,1}, ..., r_{j,n_{qb}}$ (auxquels on a appliqué $x \to \frac{1}{2} + \frac{atan(x)}{\pi}$) et les $h_{n_{qb}+1}, ..., h_{2n_{qb}}$ correspondent aux $\theta_{j,1}, ..., \theta_{j,n_{qb}}$. La fonction de paramétrisation $\boldsymbol{f}$ n'est pas parfaite non plus (et on ne peut rien obtenir de parfait, il n'existe pas de bijection dérivable d'un fermé dans un ouvert) : si le minimum que l'on cherche est atteint pour un état $\boldsymbol{v}_j$ dont la décomposition en produit tensoriel de qubits contient un qubit qui vaut $|0\rangle$ ou $|1\rangle$ (ou un $r_{j,n_{qb}}$ qui vaut 0 ou 1), alors la minimisation sera plus compliquée, car les paramètres $\{h_1, ..., h_{2n_{qb}}\}$ qui correspondent auront un élément infini. En pratique, ce n'est pas vraiment un problème, même si la valeur du qubit est exactement $|0\rangle$ ou $|1\rangle$ (probabilité nulle), l'algorithme d'optimisation va quand même converger (en plus de temps) vers un vecteur dont une composante sera très grande (autours de 10^{15}). Mathématiquement, cette solution n'est pas exacte, mais numériquement, le logiciel de calcul qui fonctionne avec des réels en virgule flottante et en double précision ne fait pas la différence quand il calcule le critère. Nous utilisons donc (3.37) pour paramétrer les états non-intriqués.

La fonction $\boldsymbol{f}_v$ est surjective (de $\mathbb{R}^{2n_{qb}}$ dans l'ensemble des états non-intriqués) et on peut définir son inverse "à droite" $\boldsymbol{f}_v^{-1}$ tel que $\boldsymbol{f}_v \circ \boldsymbol{f}_v^{-1}$ vaut l'identité ($\boldsymbol{f}_v^{-1}$ sera utile dans l'algorithme de la section 3.3.4) qui prend un vecteur non-intriqué $\boldsymbol{v}$ en entrée et retourne des paramètres $h_1, ..., h_{2n_{qb}}$ en sortie tels que $\boldsymbol{v} = \boldsymbol{f}_v(h_1, ..., h_{2n_{qb}})$. Toute fonction surjective a une inverse à droite. On étend cette définition aux vecteurs intriqués en calculant d'abord le vecteur non intriqué le plus proche (norme de Frobenius) du vecteur intriqué d'entrée (par optimisation) puis en lui appliquant l'inverse à droite.

$\mathbf{M}$ est une matrice unitaire, ses valeurs propres sont donc de module unitaire et ses espaces propres sont donc orthogonaux (montré dans Annexe B.3.3). Il existe donc une base orthonormale (représentée par une matrice unitaire $\mathbf{P}$) qui ne contient que des vecteurs propres de $\mathbf{M}$, et la décomposition en valeurs propres de $\mathbf{M}$ peut s'écrire :

$$\mathbf{M} = \mathbf{P} \begin{pmatrix} e^{i\lambda_1} & & \\ & \ddots & \\ & & e^{i\lambda_d} \end{pmatrix} \mathbf{P}^* = \mathbf{P} \exp(i\mathbf{\Lambda})\mathbf{P}^* = \exp(i\mathbf{P}\mathbf{\Lambda}\mathbf{P}^*)$$

où $\mathbf{\Lambda}$ est la matrice diagonale qui contient les λ_j sur la diagonale et exp est l'exponentielle matricielle (définie avec la décomposition en série entière, elle préserve les vecteurs propres et transforme les valeurs propres en leurs exponentielles scalaires). On remarque que, comme $\mathbf{P}$ est unitaire, la matrice $\mathbf{P}\mathbf{\Lambda}\mathbf{P}^*$ est hermitienne. Nous avons donc montré qu'il existe une surjection de l'ensemble des matrices hermitiennes dans l'ensemble des matrices unitaires :

$$\begin{cases} \mathbb{H}_d(\mathbb{C}) & \longrightarrow & \mathbb{U}_d(\mathbb{C}) \\ \mathbf{\Sigma} & \longrightarrow & \exp(i\mathbf{\Sigma}) \end{cases}.$$

Nous aurions pu créer une bijection si nous avions défini l'ensemble de départ comme les matrices hermitiennes dont toutes les valeurs propres dans $[0, 2\pi[$, mais nous n'avons besoin que d'une surjection. Cette fonction est utile car les matrices hermitiennes peuvent être directement paramétrées avec des valeurs réelles, toute matrice hermitienne s'écrit :

$$\mathbf{H}(h_1, ..., h_{d^2}) = \begin{pmatrix} h_1 & h_2 & h_4 & h_7 & ... & h_{d(d-1)/2+1} \\ h_2 & h_3 & h_5 & h_8 & ... & h_{d(d-1)/2+2} \\ h_4 & h_5 & h_6 & h_9 & ... & h_{d(d-1)/2+3} \\ h_7 & h_8 & h_9 & h_{10} & ... & h_{d(d-1)/2+4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{d(d-1)/2+1} & h_{d(d-1)/2+2} & h_{d(d-1)/2+3} & h_{d(d-1)/2+4} & ... & h_{d(d+1)/2} \end{pmatrix}$$

$$+i \begin{pmatrix} 0 & h_{d(d+1)/2+1} & h_{d(d+1)/2+2} & h_{d(d+1)/2+4} & ... & h_{d(d-1)+2} \\ -h_{d(d+1)/2+1} & 0 & h_{d(d+1)/2+3} & h_{d(d+1)/2+5} & ... & h_{d(d-1)+3} \\ -h_{d(d+1)/2+2} & -h_{d(d+1)/2+3} & 0 & h_{d(d+1)/2+6} & ... & h_{d(d-1)+4} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -h_{(d-1)^2+3} & -h_{(d-1)^2+4} & ... & -h_{d(d-1)+1} & 0 & h_{d^2} \\ -h_{d(d-1)+2} & -h_{d(d-1)+3} & -h_{d(d-1)+4} & ... & -h_{d^2} & 0 \end{pmatrix} \tag{3.38}$$

En utilisant cette paramétrisation et la surjection des matrices hermitiennes vers les matrices unitaires, on peut créer une fonction $\mathbf{F}^0_{\mathbf{M}}$ qui peut paramétrer toute matrice unitaire avec d^2 paramètres réels $\mathbf{F}^0_{\mathbf{M}}(h_1, ..., h_{d^2}) \longrightarrow \exp(i\mathbf{H}(h_1, ..., h_{d^2}))$. Cependant, nous ne voulons paramétrer $\mathbf{M}$ qu'à une phase globale près. Il se trouve qu'avec la paramétrisation que nous avons choisie, deux matrices unitaires sont les mêmes à une phase près si tous leurs paramètres sont les mêmes sauf les paramètres de la diagonale $(h_1, h_3, ..., h_{d(d+1)/2})$ qui, modulo 2π, sont les mêmes à une constante additive près (car $\exp(i(\mathbf{H} + \lambda\mathbf{I}_d)) = e^{i\lambda}\exp(i\mathbf{H})$).

Donc si on ne veut paramétriser une matrice unitaire $\mathbf{M}$ qu'à une phase globale près, on peut soustraire h_1 à tous les éléments paramètres de la diagonale et ainsi considérer $d^2 - 1$ paramètres à la place de d^2 :

$$\mathbf{F}_{\mathbf{M}}(h_1, ..., h_{d^2-1}) \longrightarrow \exp(i\mathbf{H}(0, h_1, ..., h_{d^2-1})) \tag{3.39}$$

tous les h_k ont été renommés en h_{k-1} pour que le premier paramètre soit h_1 et pas h_2.

On pourrait montrer que la paramétrisation n'est pas injective, mais on peut quand même définir une fonction "inverse à droite" qui associe des paramètres à une matrice unitaire : $\mathbf{F}^{-1}_{\mathbf{M}}$.

Les paramètres sont calculés avec le logarithme matriciel. On dit que $\mathbf{F}_{\mathbf{M}}^{-1}$ est l'"inverse à droite de" $\mathbf{F}_{\mathbf{M}}$ car $\mathbf{F}_{\mathbf{M}} \circ \mathbf{F}_{\mathbf{M}}^{-1}$ est l'identité à une phase près.

Avec les paramétrisations de (3.37) et (3.39), le problème de (3.34) devient :

$$\underset{h_1,...,h_{d^2-1}}{\arg\min} \sum_{j=1}^{n_i} \min_{h_1^j,...,h_{2n_{qb}}^j} \sum_{k=1}^{n_s} \mathcal{L}^s_{\mathbf{F}_{\mathbf{M}}(h_1,...,h_{d^2-1})^k \times \boldsymbol{f_v}(h_1^j,...,h_{2n_{qb}}^j)}(\mathbf{n}_{j,k}). \tag{3.40}$$

3.3.4 Optimisation

Le problème d'optimisation (3.40) se résout efficacement par minimisation successive :

1. On initialise $\widehat{\mathbf{M}}$ à $\widehat{\mathbf{M}}_{LS}$ et $\widehat{\boldsymbol{h}}_{\mathbf{M}}$ à $\boldsymbol{f}_{\mathbf{M}}^{-1}(\widehat{\mathbf{M}})$.
2. On initialise $\begin{bmatrix}\widehat{\boldsymbol{v}}_1 & ... & \widehat{\boldsymbol{v}}_{n_{qb}}\end{bmatrix}$ aux estimées de (3.30) et $\begin{bmatrix}\widehat{\boldsymbol{h}}_1 & ... & \widehat{\boldsymbol{h}}_{n_{qb}}\end{bmatrix}$ à $\begin{bmatrix}\boldsymbol{f}_{\boldsymbol{v}}^{-1}(\widehat{\boldsymbol{v}}_1) & ... & \boldsymbol{f}_{\boldsymbol{v}}^{-1}(\widehat{\boldsymbol{v}}_{n_{qb}})\end{bmatrix}$.
3. Pour j allant de 1 à n_s :
   (a) On résout $(\widehat{h}_1^j,...,\widehat{h}_{2n_{qb}}^j) = \underset{h_1^j,...,h_{2n_{qb}}^j}{\arg\min} \mathcal{L}^s_{\widehat{\mathbf{M}}^k \times \boldsymbol{f_v}(h_1^j,...,h_{2n_{qb}}^j)}(\mathbf{n}_{j,k})$.
   (b) $\widehat{\boldsymbol{h}_j} \leftarrow [\widehat{h}_1^j,...,\widehat{h}_{2n_{qb}}^j]^T$.
   (c) $\widehat{\boldsymbol{v}_j} \leftarrow \boldsymbol{f_v}(\widehat{h}_1^j,...,\widehat{h}_{2n_{qb}}^j)$.
4. Une itération de $(\widehat{h}_1,...,\widehat{h}_{d^2-1}) = \underset{h_1,...,h_{d^2-1}}{\arg\min} \sum_{j=1}^{n_i} \sum_{k=1}^{n_s} \mathcal{L}^s_{\mathbf{F}_{\mathbf{M}}(h_1,...,h_{d^2-1})^k \times \widehat{\boldsymbol{v}_j}}(\mathbf{n}_{j,k})$ initialisée à $\widehat{\boldsymbol{h}}_{\mathbf{M}}$
5. Si $||\widehat{\boldsymbol{h}}_{\mathbf{M}} - \begin{bmatrix}\widehat{h}_1 & ... & \widehat{h}_{d^2-1}\end{bmatrix}^T|| < 10^{-30}$ ou si on est passé par l'étape 4 plus de 700 fois (empirique), on sort de l'algorithme, les résultats dont les $\widehat{h}_1,...,\widehat{h}_{d^2-1}$ (paramètres de $\mathbf{M}$).
6. $\widehat{\boldsymbol{h}}_{\mathbf{M}} \leftarrow \begin{bmatrix}\widehat{h}_1 & ... & \widehat{h}_{d^2-1}\end{bmatrix}^T$ et $\widehat{\mathbf{M}} \leftarrow \mathbf{F}_{\mathbf{M}}(\widehat{h}_1,...,\widehat{h}_{d^2-1})$.
7. On remplace les $\begin{bmatrix}\widehat{\boldsymbol{v}}_1 & ... & \widehat{\boldsymbol{v}}_{n_{qb}}\end{bmatrix}$ par les estimées de l'équation (3.30) et $\begin{bmatrix}\widehat{\boldsymbol{h}}_1 & ... & \widehat{\boldsymbol{h}}_{n_{qb}}\end{bmatrix} \leftarrow \begin{bmatrix}\boldsymbol{f}_{\boldsymbol{v}}^{-1}(\widehat{\boldsymbol{v}}_1) & ... & \boldsymbol{f}_{\boldsymbol{v}}^{-1}(\widehat{\boldsymbol{v}}_{n_{qb}})\end{bmatrix}$.
8. Aller à l'étape 3.

Il convient de clarifier les étapes 3a et 4 :

- Pour l'étape 3a, on minimise $\mathcal{L}^s_{\widehat{\mathbf{M}}^k \times \boldsymbol{f_v}(h_1^j,...,h_{2n_{qb}}^j)}(\mathbf{n}_{j,k})$ par rapport à $(h_1^j,...,h_{2n_{qb}}^j)$ avec un algorithme quasi-Newton qui estime la hessienne avec la méthode BFGS [Bro70] (fonction fminunc en MATLAB) initialisée à $\widehat{\boldsymbol{h}_j}$. L'algorithme BFGS a besoin de pouvoir calculer les gradients à chaque étape, pour ce faire, on a créé des fonctions MATLAB qui calculent les gradients théoriques (en multipliant des jacobiennes). Avec ces gradients, l'algorithme BFGS met à jour son estimée de la hessienne et calcule une direction dans laquelle MATLAB fait une recherche linéaire du minimum (qui est le prochain point de l'algorithme). En général, la direction n'est pas exactement la direction du minimum (car la hessienne n'est pas exacte et la fonction à minimiser n'est pas d'ordre 2), mais le critère a forcément diminué. L'algorithme s'arrête si l'une des trois conditions suivantes est satisfaite : (i) le critère a été évalué 300 fois. (ii) Le critère a changé de moins de 10^{-8} (valeur absolue). (iii) La norme du changement du vecteur des paramètres vaut moins de 10^{-7}.

- Pour l'étape 4 on utilise le même algorithme, mais on ne fait qu'une seule itération, et l'estimée de la hessienne que l'on avait la dernière fois que l'on est passé à l'étape 4 est mise à jour (l'estimée de la hessienne est initialisée à l'identité).

Le code pour calculer les gradients de façon analytique a demandé beaucoup de travail (les critères sont complexes et certaines fonctions comme l'exponentielle matricielle sont difficiles à dériver), mais pour un algorithme de descente de gradient, la seule alternative est d'estimer les gradients par différences finies. Cela prendrait beaucoup plus de temps que notre solution (il y a $O(2^{2n_{qb}})$ paramètres à optimiser donc $O(2^{2n_{qb}})$ directions à observer) et rendrait l'algorithme 10 fois plus lent pour deux qubits et irréalisable pour plus de deux qubits.

Les limites sur le nombre d'itérations (300 pour les vecteurs initiaux et 700 pour la matrice de processus) sont très rarement atteintes, sauf pour 4 ou 5 qubits (on ne dépasse pas 5 qubits). Dans ce cas, l'optimisation des états initiaux converge toujours en moins de 300 itérations (on n'a que $2n_{qb}$ paramètres à trouver), mais l'optimisation de $\mathbf{M}$ peut nécessiter plus de 700 itérations. Nous avons posé la limite de 700 itérations pour que l'algorithme soit plus rapide, en pratique, les premières itérations changent beaucoup les paramètres et diminuent beaucoup le critère, mais, après (à peu près) 500 itérations, le progrès est faible.

3.3.5 Borne de Cramér-Rao

Un des avantages d'avoir défini la vraisemblance et d'avoir calculé ses dérivées en fonction des paramètres est que l'on peut calculer la borne de Cramér-Rao qui est une limite inférieure pour la matrice de covariance de tout estimateur non-biaisé. On définit le vecteur des paramètres $\boldsymbol{\theta} = \begin{pmatrix} h_1 & ... & h_{d^2-1} & h_1^1 & ... & h_{2n_{qb}}^{n_i} \end{pmatrix}^T$ de taille $n_\theta = d^2 + 2n_{qb}n_i - 1$. C'est par rapport à ces paramètres que la vraisemblance des mesures est maximisée (même si on a choisi de le faire par optimisation successive). La borne de Cramér-Rao est définie comme l'inverse de l'information de Fisher $\mathbf{I}^s(\boldsymbol{\theta}_0)$ ($\boldsymbol{\theta}_0$ est la vraie valeur des paramètres) qui est définie comme la matrice carrée de taille n_θ dont l'élément en ligne j et colonne k vaut :

$$\left(\mathbf{I}^s(\boldsymbol{\theta}_0)\right)_{j,k} = -\mathbb{E}_{\theta_0}\left(\frac{\partial^2 \mathcal{L}^s_{\boldsymbol{\theta}}(\mathbf{N})}{\partial\theta_j \partial\theta_k}(\boldsymbol{\theta}_0)\right) \tag{3.41}$$

où $\mathbf{N}$ est vu comme une variable aléatoire dont $\mathbb{E}_{\theta_0}$ est l'espérance selon le modèle multinomial (si $s = exact$) ou gaussien (si $s = gauss$) dont les paramètres sont fonctions de θ pour $\theta = \theta_0$, et $\mathcal{L}^s_{\boldsymbol{\theta}}(\mathbf{N})$ est la log-vraisemblance négative de $\mathbf{N}$ que l'on avait définie dans (3.33) sous la forme $\mathcal{L}^s_{\mathbf{V},\mathbf{M}}(\mathbf{N})$. Ici, $\mathbf{M}$ est calculée avec les $d^2 - 1$ premiers paramètres de θ et $\mathbf{V}$ est calculée en fonction des $n_{qb}n_i$ derniers.

L'annexe B.4 montre que les expressions de l'information de Fisher pour les deux types de vraisemblance que nous considérons sont les suivantes :

$$\begin{aligned} \mathbf{I}^{exact}(\boldsymbol{\theta}_0) &= n_c \mathbf{J}(\boldsymbol{\theta}_0)^* \begin{pmatrix} p_1(\boldsymbol{\theta}_0) & & \\ & \ddots & \\ & & p_{n_{prob}}(\boldsymbol{\theta}_0) \end{pmatrix}^{-1} \mathbf{J}(\boldsymbol{\theta}_0) \\ \mathbf{I}^{gauss}(\boldsymbol{\theta}_0) &= n_c \mathbf{J}(\boldsymbol{\theta}_0)^* \begin{pmatrix} \widetilde{p}_1(\boldsymbol{\theta}_0) & & \\ & \ddots & \\ & & \widetilde{p}_{n_{prob}}(\boldsymbol{\theta}_0) \end{pmatrix}^{-1} \mathbf{J}(\boldsymbol{\theta}_0) \end{aligned} \tag{3.42}$$

où $p_1(\boldsymbol{\theta}_0) = \mathbb{E}_{\theta_0}\left(\frac{(\boldsymbol{n}_{1,1})_1}{n_c}\right)$ $p_2(\boldsymbol{\theta}_0) = \mathbb{E}_{\theta_0}\left(\frac{(\boldsymbol{n}_{1,1})_2}{n_c}\right), ..., p_{n_{prob}}(\boldsymbol{\theta}_0) = \mathbb{E}_{\theta_0}\left(\frac{(\boldsymbol{n}_{n_i,n_s})_d}{n_c}\right)$ sont les valeurs théoriques des probabilités mesurées ($(\boldsymbol{n}_{1,1})_k$ est le k ème élément de $\boldsymbol{n}_{1,1}$), la matrice $\mathbf{J}(\boldsymbol{\theta}_0)$ est la jacobienne qui contient les dérivées partielles de ces probabilités en fonction des paramètres

θ. Elle est de taille $n_\theta \times n_{prob}$, $n_{prob} = dn_i n_s n_t$, et $\widetilde{p}_1(\boldsymbol{\theta}_0) = \mathbb{E}_{\theta_0}\left(\frac{(\boldsymbol{n}_{1,1})_1+5}{n_c+5d}\right), ..., \widetilde{p}_{n_{prob}}(\boldsymbol{\theta}_0) = \mathbb{E}_{\theta_0}\left(\frac{(\boldsymbol{n}_{n_i,n_s})_d+5}{n_c+5d}\right)$ sont les espérances des probabilités régularisées.

La borne de Cramèr Rao dépend des vrais paramètres θ_0. En pratique, ils sont inconnus, on doit donc les remplacer par les paramètres estimés par maximum de vraisemblance.

En pratique, ce n'est pas la matrice de covariance $\boldsymbol{\theta}$ estimée qui nous intéresse, mais la matrice de covariance de la $\widehat{\mathbf{M}}_{ML}$ de rotation que l'on estime à partir des d^2-1 paramètres de $\widehat{\mathbf{M}}_{ML}$. On peut facilement passer de l'une à l'autre avec la jacobienne de $\boldsymbol{f}_{vect} \circ \mathbf{F}_{\mathbf{M}} : \mathbf{J}_{\mathbf{m}}(\theta_0)$ de taille $2d^2 \times (d^2-1)$, où $\boldsymbol{f}_{vect}$ est la fonction qui prend une matrice complexe $\mathbf{M}$ de taille d en entrée et rend en sortie le vecteur $\boldsymbol{m}$ de taille $2d^2$ qui contient les concaténations verticales des parties réelles et parties imaginaires des colonnes de $\mathbf{M}$ les unes sous les autres :

$$\boldsymbol{\Sigma}^0_{\widehat{\boldsymbol{m}}}(\theta_0) = \mathbf{J}_{\mathbf{M}}(\theta_0)\boldsymbol{\Sigma}^{cr}_{\mathbf{h}}(\theta_0)\mathbf{J}^*_{\mathbf{M}}(\theta_0) \tag{3.43}$$

où $\boldsymbol{\Sigma}^0_{\widehat{\boldsymbol{m}}}(\theta_0)$ est la borne inférieure de la matrice de covariance (hermitienne complexe) des parties réelles et imaginaires des éléments de la matrice de rotation estimée, et $\boldsymbol{\Sigma}^{cr}_{\mathbf{h}}(\theta_0)$ contient le bloc $(d^2-1) \times (d^2-1)$ en haut à gauche de la borne de Cramér-Rao (inverse de l'information de Fisher de (3.41)).

En pratique, cette estimée peut donner une fausse impression de l'erreur car, même si la fonction $\mathbf{F}_{\mathbf{M}}$ a été construite de façon à ce que des petites variations des entrées ne puissent pas changer la matrice unitaire en la multipliant par un facteur de phase global (en fixant le premier paramètre à 0), il peut quand même exister des petites variations des paramètres qui génèrent une erreur proche d'une phase globale. Par exemple, pour $d = 4$

$$\mathbf{M}_1 = \mathbf{F}_{\mathbf{M}}(0, ..., 0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{M}_2 = \mathbf{F}_{\mathbf{M}}(0, \epsilon, 0, 0, \epsilon, 0, 0, 0, \epsilon, 0, ..., 0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\epsilon} & 0 & 0 \\ 0 & 0 & e^{i\epsilon} & 0 \\ 0 & 0 & 0 & e^{i\epsilon} \end{pmatrix}.$$

L'erreur coefficient par coefficient entre les deux matrices est $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1-e^{i\epsilon} & 0 & 0 \\ 0 & 0 & 1-e^{i\epsilon} & 0 \\ 0 & 0 & 0 & 1-e^{i\epsilon} \end{pmatrix}$ mais on peut avoir une erreur de norme plus petite en multipliant $\mathbf{M}_2$ par un bon facteur de phase comme $e^{-i\epsilon}$ (on peut montrer que le facteur de phase qui minimise la norme de l'erreur est $\frac{tr(\mathbf{M}_1\mathbf{M}_2^*)}{|tr(\mathbf{M}_1\mathbf{M}_2^*)|}$). Pour remédier à cette surestimation de l'erreur, nous multiplions les matrices $\widehat{\mathbf{M}}$ en sortie par le facteur de phase $\frac{tr(\mathbf{M}_0\widehat{\mathbf{M}}^*)}{|tr(\mathbf{M}_0\widehat{\mathbf{M}}^*)|}$ où $\mathbf{M}_0$ est une matrice que l'on choisit à l'avance. En simulation, on peut prendre la vraie matrice du processus, mais en pratique, on peut prendre l'identité.

On peut exprimer la borne inférieure de la matrice de covariance de $\boldsymbol{f}_{vect}\left(\frac{tr(\mathbf{M}_0\widehat{\mathbf{M}}^*)}{|tr(\mathbf{M}_0\widehat{\mathbf{M}}^*)|}\widehat{\mathbf{M}}\right)$:

$$\boldsymbol{\Sigma}_{\widehat{\mathbf{M}}}(\boldsymbol{\theta}_0, \mathbf{M}_0) = \mathbf{J}_{reph}(\boldsymbol{\theta}_0, \mathbf{M}_0)\mathbf{J}_{\mathbf{m}}(\boldsymbol{\theta}_0)\boldsymbol{\Sigma}^{cr}_{\mathbf{h}}(\boldsymbol{\theta}_0)\mathbf{J}_{\mathbf{m}}(\boldsymbol{\theta}_0)^*\mathbf{J}_{reph}(\boldsymbol{\theta}_0, \mathbf{M}_0)^* \tag{3.44}$$

où $\mathbf{J}_{reph}(\boldsymbol{\theta}_0, \mathbf{M}_0)$ est la jacobienne de $\mathbf{F}_{rephasage} : \boldsymbol{m} \to \boldsymbol{f}_{vect}\left(\frac{tr(\mathbf{M}_0\boldsymbol{f}^{-1}_{vect}(\boldsymbol{m})^*)}{|tr(\mathbf{M}_0\boldsymbol{f}^{-1}_{vect}(\boldsymbol{m})^*)|}\boldsymbol{f}^{-1}_{vect}(\boldsymbol{m})\right)$ avec la convention que $\boldsymbol{f}^{-1}_{vect}$ est la fonction inverse de $\boldsymbol{f}_{vect}$. La jacobienne est calculée en le point $\mathbf{m}$ associé à $\boldsymbol{\theta}_0$ ($\mathbf{m} = (\boldsymbol{f}_{vect} \circ \mathbf{F}_{\mathbf{M}})((\boldsymbol{\theta}_0)_1, ..., (\boldsymbol{\theta}_0)_{d^2-1})$) et $\mathbf{M}_0$ est une matrice donnée considérée comme constante pour le calcul de la jacobienne. Nous devons considérer les parties réelles et imaginaires de l'erreur sur la matrice de rotation, car $\mathbf{F}_{rephasage}$ n'est pas dérivable en fonction des coefficients complexes (à cause des transconjuguées) mais elle est dérivable en fonction des coefficients réels si la diagonale de $\mathbf{M}_0\boldsymbol{f}^{-1}_{vect}(\boldsymbol{m})^*$ ne contient aucun 0.

Nous considérons que $\boldsymbol{\Sigma}_{\widehat{\mathbf{M}}}(\boldsymbol{\theta}_0, \mathbf{M}_0)$ est la seule estimée de la covariance de l'erreur qui vaut la peine d'être calculée. En pratique on ne connait pas $\boldsymbol{\theta}_0$, on le remplace donc par $\widehat{\boldsymbol{\theta}}_{ML}$ qui est l'estimée des paramètres en sortie de l'algorithme de maximum de vraisemblance. $\mathbf{M}_0$ est arbitraire, on peut prendre $\widehat{\mathbf{M}}_{ML}$ pour s'assurer que tous les coefficients de $\mathbf{M}_0 \boldsymbol{f}_{vect}^{-1}(\boldsymbol{m})^*$ soient loin de 0.

3.4 Conclusion

Nous avons abordé les problèmes de QPT et QST de façon similaire : nous avons un algorithme initial rapide (section 3.1) et peu précis qui peut être amélioré par un algorithme de maximum de vraisemblance (section 3.3). Nous voyons ces algorithmes comme une contribution significative car :

- Ils fonctionnent avec des états d'entrée qui sont (a priori) quelconques. Cela nous permet d'introduire la configuration semi-aveugle.
- Nous avons pu montrer que si la configuration (valeur des états initiaux) rend la QPT possible, alors, notre algorithme initial permet toujours de trouver $\mathbf{M}$ s'il n'y a pas d'erreur de QST.
- Les deux algorithmes devraient (et nous allons le vérifier) être résistants aux erreurs (erreurs de QPT pour l'algorithme initial et erreurs sur les probabilités empiriques des résultat pour l'algorithme de ML), car ils ont été pensés pour minimiser des métriques pertinentes (normes de l'erreur de QST pour l'un, vraisemblance pour l'autre).
- L'algorithme de ML nous donne une estimée de la covariance de l'erreur (borne de Cramér-Rao).

Comme la QPT est l'objectif principal de la thèse, les performances des algorithmes seront testées dans le chapitre 5 qui leur est dédié.

Chapitre 4

Tomographie de mesures aveugle

Sommaire

4.1 Objectifs

Nos algorithmes de QPT introduits dans le chapitre 3 font les hypothèses suivantes sur le système :

1. Les états d'entrée sont purs et non-intriqués.
2. Le processus à identifier est unitaire.
3. Les mesures que l'on effectue suivent le modèle des mesures projectives de la section 1.1.4 et on connait exactement la matrice des vecteurs propres qui représente chaque type de mesures.

Nous pourrions potentiellement nous passer de la première hypothèse. Les états d'entrée auraient très bien pu être intriqués, mais les états non-intriqués sont plus faciles à préparer, et la paramétrisation des états non-intriqués est beaucoup plus simple que la paramétrisation d'états purs quelconques ($2n_{qb}$ paramètres contre $2d-2$) ce qui rend la maximisation de la vraisemblance pour la QPT (section 3.3) plus simple. Par ailleurs, si les états d'entrée sont des états mélange, mais non dégénérés (toutes les valeurs propres de la matrice densité de chaque état sont distinctes), alors nous pouvons les estimer en faisant de la tomographie d'état mélange plutôt que de la tomographie d'état pur. Nos algorithmes peuvent facilement être adaptés à des états mélanges, car un opérateur unitaire agit sur les vecteurs propres d'un état mixte de la même façon qu'il agit sur le vecteur qui représente un état pur. Nous choisissons de rester sur des états purs, car (i) ils ont moins de paramètres, et peuvent être identifiés avec moins de mesures. (ii) Ils sont une bonne approximation des états générés par un ordinateur quantique. En effet, (presque) tous les états que l'on cherche à générer sont purs, les imperfections du "hardware"

les transforment en états mélanges mais ils ne s'éloignent pas trop du modèle. (iii) Si les portes utilisées pour préparer les états sont bien unitaires (hypothèse qu'il est raisonnable de faire sur ces portes car on la fait sur la porte à identifier) et qu'elles agissent bien sur un état premier dont tous les qubits sont initialisés à $|0\rangle$ (c'est la base de l'informatique quantique), alors les états présentés en entrée du processus à identifier sont forcément purs car les portes unitaires préservent le caractère pur des états. (iv) Cette hypothèse est significativement plus raisonnable que l'hypothèse qui est généralement faite pour la QPT unitaire : "l'opérateur est capable de préparer les états d'entrée à des valeurs prédéterminées".

La deuxième hypothèse est centrale dans tous nos algorithmes et nous ne pouvons pas nous en passer. Les modèles de processus non-unitaires ont trop de paramètres pour que notre algorithme de maximum de vraisemblance (section 3.3.1) soit réaliste. De plus, notre algorithme de moindres carrés totaux (section 3.1.2) devient presque trivial pour les processus non-unitaire car les mesures dépendent des d^4 paramètres de la matrice de processus $\boldsymbol{\chi}$ de façon linéaire, voir (1.14), et exploiter cette relation pour estimer le processus n'est pas du tout une nouvelle idée (voir [CN97]). Nous défendons cette hypothèse dans la section 1.12.1.

La troisième hypothèse sera partiellement remise en question dans le présent chapitre. En réalité, l'hypothèse que nous avons implicitement faite sur les mesures est légèrement plus restrictives que l'hypothèse 3 : nous supposons que les mesures mono-qubit X, Y et Z sont exactement conformes au modèle. C'est-à-dire que nous ne supposons pas seulement que les paramètres de la mesure sont connus, mais aussi qu'ils sont exactement définis par les matrices de (2.2), que l'on rappelle ici :

$$\mathbf{E}_X = \mathbf{U}_k(\tfrac{\pi}{4}, \pi) = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{E}_Y = \mathbf{U}_k(\tfrac{\pi}{4}, \tfrac{\pi}{2}) = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}, \quad \mathbf{E}_Z = \mathbf{U}_k(0,0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.1}$$

où $\mathbf{U}_k(\phi, \psi)$ défini en (2.1) paramétrise les matrices unitaires dont les lignes sont définies à une phase près : $\mathbf{U}_k(\phi, \psi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi)e^{i\psi} \\ \sin(\phi) & \cos(\phi)e^{i\psi} \end{pmatrix}$.

Cependant, en pratique, si les mesures ne sont pas exactement représentées par les matrices de (4.1), mais que nous connaissons les matrices de vecteurs propres qui les caractérisent, alors nous pouvons facilement adapter nos algorithmes pour qu'ils fonctionnent quand même. Les algorithmes phaseCut (section 2.2) et de maximum de vraisemblance pour la QST (section 2.4) et pour la QPT (section 3.3) ont comme entrée les paramètres des mesures (nécessaires pour connaître la matrice $\mathbf{A}$[1]) mais les valeurs de ces paramètres ne sont pas prédéterminées : s'ils ne sont pas exactement les paramètres de (4.1), mais sont un peu différents, les algorithmes marcheront tout aussi bien tant que les matrices de vecteurs propres (générées par les paramètres) sont parfaitement connues. On retrouve donc simplement l'hypothèse 3 : les valeurs des paramètres des mesures doivent être connues. L'algorithme récursif de la section 2.3 est une exception. Il ne peut pas être adapté si les types de mesures changent et restent connus. Cependant, cet algorithme n'est utilisé que pour obtenir une estimée "pas trop mauvaise" des états mesurés qui sera ensuite raffinée avec l'algorithme de QST par maximum de vraisemblance de la section 2.4. Tant que les mesures sont assez proches du modèle, il est raisonnable de supposer que l'estimée de l'algorithme récursif est suffisamment bonne pour que l'algorithme de vraisemblance converge vers le bon minimum. C'est d'autant plus vrai que nous avons vu (dans la section 2.5.4) que en pratique, avec les mesures de la section 2.3, l'algorithme de QST par maximum de vraisemblance fonctionne même avec des mesures aléatoires (il est juste plus lent qu'avec l'initialisation au résultat de l'algorithme récursif). C'est pour ces raisons que nous avons formulé l'Hypothèse

[1] On rappelle que $\mathbf{A}$ est la concaténation verticale des matrices de vecteurs propres des types de mesures effectués. Voir les sections 2.2 et 2.3.

3 ainsi et n'avons pas imposé aux mesures d'être conformes au modèle de (4.1).

Même sous cette forme plus faible, l'Hypothèse 3 est moins facile à justifier que les deux autres. Cette thèse est partie de l'idée qu'il est trop contraignant de supposer que les états que l'opérateur est capable de préparer soient connus avant l'expérience car ils sont préparés avec des portes quantiques (des portes mono-qubit dans notre cas car les états sont non-intriqués). Or, pour certaines architectures d'ordinateurs quantiques, les mesures ne sont possibles que dans la base de référence, et, pour réaliser tout autre type de mesure, on doit appliquer une porte unitaire à l'état que l'on veut mesurer (ex : porte de Hadamard pour faire la mesure X) avant de faire une mesure dans la base de référence. Le fait d'affirmer que ces portes mono-qubit peuvent être considérées comme fiables, mais que ce n'est pas le cas des portes mono-qubit qui servent à préparer les états d'entrée peut sembler hypocrite. Cependant, on pourrait défendre le travail réalisé jusqu'ici dans la présente thèse de deux manières différentes :

- Les mesures peuvent être vues comme une référence (voir section suivante). On peut par exemple définir l'axe $(10...0)^T$ (taille d) de l'espace de Hilbert à une phase près, comme l'ensemble des états qui donnent toujours $0, .., 0$ (n_{qb} fois) quand on les mesure dans la base $Z...Z$. Idem pour $(010...0)^T$ avec le résultat $10...0$ etc. Avec cette définition, dire que "le type de mesure $Z...Z$ n'est pas conforme au modèle" n'a pas de sens car la matrice de vecteurs propres associée à la mesure ne peut pas être fausse, elle est forcément orthogonale, et elle définit le choix des axes de la base de référence. Par contre, on ne peut utiliser cet argument qu'une fois : les mesures ou les états d'entrée peuvent servir de références (en partie, voir section suivante) pour les axes de l'espace de Hilbert, mais pas les deux. C'est un argument pour défendre notre choix de faire confiance aux mesures mais pas aux états d'entrée, nous aurions pu faire l'inverse, mais il fallait choisir l'un ou l'autre. Nous verrons dans la section suivante que cet argument ne peut pas être utilisé pour affirmer qu'il existe une base de référence dans laquelle toutes les mesures sont parfaites, mais on peut trouver une base qui rend certaines composantes d'erreurs nulles.

- Même si la qualité de l'estimée de $\mathbf{M}$ est limitée par la qualité des mesures, le choix que nous avons fait de n'utiliser que des mesures non-intriquées fait que les mesures sont plus faciles à réaliser expérimentalement. Il est vrai que, de la même manière, nous aurions pu argumenter que, comme les états d'entrée sont non-intriqués, ils peuvent être préparés de manière précise. Cependant, avant nos contributions, il n'existait pas (dans la littérature) d'algorithme de tomographie de processus spécifiquement adapté aux processus unitaires qui n'utilisent que des mesures et des états non-intriquées (à part pour un seul qubit).

Nous avons jugé ces arguments insuffisants et nous avons travaillé sur la tomographie de mesures (QMT "quantum measurement tomography"). Nous ne sommes pas les premiers à étudier ce problème, il est parfois appellé "quantum detector tomography" dans la littérature, il a été étudié sous une forme non-aveugle en mesurant des états connus avec les mesures à identifier, et aveugle en estimant simultanément les états mesurés et les mesures effectuées (voir section 1.3). Pour rester cohérent avec la tomographie de processus, nous devons considérer que les états mesurés sont inconnus. Mais les mesures que nous effectuons ont des particularités qui font que les algorithmes de QMT de la littérature ne sont pas adaptés : (i) nous n'utilisons que des mesures non-intriquées (i.e. séparables en mesures mono-qubit) (ii) sur chaque qubit, nous ne faisons que 3 types de mesures différents.

L'algorithme de QMT (quantum measurement tomography) que nous allons définir dans le présent chapitre est adapté aux trois types de mesures (X, Y et Z) que nous effectuons. Nos hypothèses sont bien plus restrictives que celles des algorithmes de QMT de la littérature. En effet, nous supposons que les mesures que l'on fait sont bien des mesures projectives et sont non-intriquées, alors que, en général, la QMT calcule tous les paramètres d'un POVM inconnu.

Cependant, les hypothèses que nous faisons font que les mesures sont traitées de la même manière que les état initiaux : nous considérons qu'elles sont bien effectuées avec des portes unitaires mono-qubit (avec la représentation de mesure par porte interposée à droite de la figure 2.1), mais nous ne faisons pas confiance aux valeurs des paramètres de ces portes.

Dans la section 4.2, nous rappelons comment les mesures sont définies et paramétrées, puis nous étudions comment la base de référence peut être définie pour minimiser le nombre de paramètres à estimer. Dans la section 4.3, nous décrirons l'algorithme qui permet de trouver les paramètres qui sont observables. Dans la section 4.4, nous verrons comment on modélise la concaténation des mesures sur plusieurs qubits, et nous expliquerons comment on peut réaliser la tomographie de mesures en même temps (avec le même circuit quantique et les mêmes mesures) que la QPT.

4.2 Mesures vues comme une référence

Nous considérons l'espace de Hilbert de dimension 2 qui représente les états avec un seul qubit. Nous allons changer la base de référence pour qu'elle "corresponde" (nous allons voir en quel sens) aux mesures que l'on réalise. C'est dans cette base de référence que seront exprimés les états estimés par la QST et le processus estimé par la QPT. La base de référence dans laquelle tous les vecteurs sont exprimés (au départ) est la base canonique en dimensions deux : $\boldsymbol{\delta}_1 = \begin{pmatrix}1\\0\end{pmatrix}$ et $\boldsymbol{\delta}_2 = \begin{pmatrix}0\\1\end{pmatrix}$. La nouvelle base de référence est définie à une phase globale près, et elle est orthogonale, on appelle les deux vecteurs qui la composent $\boldsymbol{e}_1$ et $\boldsymbol{e}_2$. Comme la base est définie à une phase près, on peut supposer que le premier élément de $\boldsymbol{e}_1$ est réel positif.

On rappelle ici l'expression générique d'une matrice de vecteurs propres associée à un type de mesure (2.1) :

$$\mathbf{E}(\phi,\psi) = \begin{pmatrix}\cos(\phi) & -\sin(\phi)e^{i\psi}\\ \sin(\phi) & \cos(\phi)e^{i\psi}\end{pmatrix} \tag{4.2}$$

où $\phi \in [0, \pi/2]$ et $\psi \in [0, 2\pi[$.

Dans la présente section, on considère les matrices de vecteurs propres dans l'ancienne base de référence $\{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2\}$: $(\mathbf{E}^1_Z = \mathbf{E}(\phi^1_z, \psi^1_z), \mathbf{E}^1_Y = \mathbf{E}(\phi^1_y, \psi^1_y), \mathbf{E}^1_X = \mathbf{E}(\phi^1_x, \psi^1_x))$, et dans la nouvelle base de référence $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$: $(\mathbf{E}^2_Z = \mathbf{E}(\phi^2_z, \psi^2_z), \mathbf{E}^2_Y = \mathbf{E}(\phi^2_y, \psi^2_y), \mathbf{E}^2_X = \mathbf{E}(\phi^2_x, \psi^2_x))$. On dit qu'une matrice de vecteurs propres est exprimée dans l'ancienne (resp. la nouvelle) base de référence quand les probabilités de la mesure associée sur un état représenté par le vecteur $\boldsymbol{v}$ sont contenues dans le module au carré du produit de la matrice de vecteurs propres en question avec le vecteur qui contient la décomposition de $\boldsymbol{v}$ dans l'ancienne (resp. la nouvelle) base de référence. Par exemple, pour Z quel que soit l'état à mesurer représenté par $\boldsymbol{v} = v^1_1\boldsymbol{\delta}_1 + v^1_2\boldsymbol{\delta}_2 = v^2_1\boldsymbol{e}_1 + v^2_2\boldsymbol{e}_2$, les probabilités de mesurer 1 et 0 quand on mesure $\boldsymbol{v}$ selon Z sont contenues dans le vecteur $\left|\mathbf{E}^1_Z \begin{pmatrix}v^1_1\\v^1_2\end{pmatrix}\right|^2$ ou dans le vecteur $\left|\mathbf{E}^2_Z \begin{pmatrix}v^2_1\\v^2_2\end{pmatrix}\right|^2$ (les deux sont égaux $\forall \boldsymbol{v}$). Les deux matrices de vecteurs propres sont liées par la relation $\mathbf{E}^2_Z = \mathbf{E}^1_Z \begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2\end{bmatrix}$, (formule de changement de base) idem pour X et Y.

Les vraies valeurs des paramètres ϕ^1_z, ψ^1_z, ϕ^1_y, ψ^1_y, ϕ^1_x et ψ^1_x sont inconnues, leurs valeurs cibles sont : 0, 0, $\pi/4$, $\pi/2$, $\pi/4$ et π respectivement, voir (2.2). En pratique, les paramètres et les valeurs cibles sont proches mais pas égaux. Les valeurs des matrices de vecteurs propres $\mathbf{E}^2_Z, \mathbf{E}^2_Y, \mathbf{E}^2_X$ et de leurs paramètres ϕ^2_z, ψ^2_z, ϕ^2_y, ψ^2_y, ϕ^2_x et ψ^2_x dépendent du choix de la nouvelle base de référence.

Notre objectif est de choisir une base de référence qui annule le plus de paramètres des mesures (ϕ_z^2, ψ_z^2, ϕ_y^2, ψ_y^2, ϕ_x^2 et ψ_x^2) possible. On choisit de fixer les vecteurs de la nouvelle base de référence $\boldsymbol{e}_1$ et $\boldsymbol{e}_2$ tels que les deux conditions suivantes soient vérifiées[2] :

1. $\exists \psi_1$ t.q. $\mathbf{E}_Z^2 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\psi_1} \end{bmatrix}$.
2. $\exists \psi_2, \psi_3$ t.q. $\mathbf{E}_X^2 = \begin{bmatrix} e^{i\psi_2} & 0 \\ 0 & e^{i\psi_3} \end{bmatrix} \mathbf{R}$ où $\mathbf{R}$ est une matrice orthogonale réelle de déterminant -1.

On rappelle que $\mathbf{E}_Z^2 = \mathbf{E}_Z^1 \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 \end{bmatrix}$ et $\mathbf{E}_X^2 = \mathbf{E}_X^1 \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 \end{bmatrix}$. Trouvons les vecteurs de base $\boldsymbol{e}_1$ $\boldsymbol{e}_2$ qui font que les deux conditions soient vérifiées.

Pour tout χ_1, $\begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 \end{bmatrix} = (\mathbf{E}_Z^1)^* \begin{pmatrix} 1 & 0 \\ 0 & e^{i\chi_1} \end{pmatrix}$ satisfait la Condition 1. Avec cette valeur de $\begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 \end{bmatrix}$ qui dépend du paramètre χ_1 à régler, calculons $\mathbf{E}_X^2$:

$\mathbf{E}_X^2 = \mathbf{E}_X^1 \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 \end{bmatrix} = \mathbf{E}_X^1 (\mathbf{E}_Z^1)^* \begin{pmatrix} 1 & 0 \\ 0 & e^{i\chi_1} \end{pmatrix}$. Écrivons la matrice unitaire $\mathbf{E}_X^1 (\mathbf{E}_Z^1)^*$ coefficient par coefficient : $\mathbf{E}_X^1 (\mathbf{E}_Z^1)^* = \begin{pmatrix} \cos(\phi_0)e^{i\psi_1} & -\sin(\phi_0)e^{i\psi_3} \\ \sin(\phi_0)e^{i\psi_2} & \cos(\phi_0)e^{i(\psi_2+\psi_3-\psi_1)} \end{pmatrix}$. Toute matrice unitaire s'écrit de cette façon (voir Annexe A.1). Ainsi, reprenons le calcul de $\mathbf{E}_X^2$:

$\mathbf{E}_X^2 = \begin{pmatrix} \cos(\phi_0)e^{i\psi_1} & -\sin(\phi_0)e^{i\psi_2} \\ \sin(\phi_0)e^{i\psi_3} & \cos(\phi_0)e^{i(\psi_2+\psi_3-\psi_1)} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\psi_1} \end{pmatrix} = \begin{pmatrix} \cos(\phi_0)e^{i\psi_1} & -\sin(\phi_0)e^{i(\psi_2+\chi_1)} \\ \sin(\phi_0)e^{i\psi_3} & \cos(\phi_0)e^{i(\psi_2+\psi_3-\psi_1+\chi_1)} \end{pmatrix}$.

On pose $\chi_1 = \psi_1 - \psi_2$ (on rappelle que la Condition 1 est vérifiée quelle que soit la valeur de χ_1), on a :

$\mathbf{E}_X^2 = \begin{pmatrix} e^{i\psi_1} & 0 \\ 0 & e^{i\psi_3} \end{pmatrix} \begin{pmatrix} \cos(\phi_0) & -\sin(\phi_0) \\ \sin(\phi_0) & \cos(\phi_0) \end{pmatrix}$. Avec cette valeur de la base de référence, on vérifie donc la Condition 2 (avec $\psi_2 = \psi_1, \psi_3 = \pi + \psi_3$).

Nous avons donc montré qu'il existe une base de référence qui fait que les Conditions 1 et 2 sont vérifiées. Cela signifie que, à des phases globales sur les lignes près (ces phases ne changent rien pour des matrices de vecteurs propres), les matrices des vecteurs propres des 3 types de mesures s'écrivent :

$$\mathbf{E}_Z^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{E}_Y^2 = \begin{pmatrix} \cos(\phi_y) & -\sin(\phi_y)e^{i\psi_y} \\ \sin(\phi_y) & \cos(\phi_y)e^{i\psi_y} \end{pmatrix}, \mathbf{E}_X^2 = \begin{pmatrix} \cos(\phi_x) & \sin(\phi_x) \\ \sin(\phi_x) & -\cos(\phi_x) \end{pmatrix} \tag{4.3}$$

ce qui réduit le nombre de paramètres :

$$\mathbf{E}_Z^2 = \mathbf{E}(0,0), \mathbf{E}_Y^2 = \mathbf{E}(\phi_y, \psi_y), \mathbf{E}_X^2 = \mathbf{E}(\phi_x, \pi). \tag{4.4}$$

Nous avons donc 3 paramètres à estimer au lieu 6. Dans tout le reste de ce chapitre, tous les vecteurs et les matrices sont exprimés dans la nouvelle base de référence $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ et les matrices de vecteurs propres que l'on considère sont $\mathbf{E}_Z = \mathbf{E}_Z^2, \mathbf{E}_Y = \mathbf{E}_Y^2, \mathbf{E}_X = \mathbf{E}_X^2$. Ce choix ne doit pas être vu comme un renoncement à estimer tous les paramètres que l'on aurait pu estimer. Sans fixer la base de référence, les 6 paramètres des mesures ne sont pas observables, la base de

[2] Nous verrons ensuite que ces conditions font que l'on a les matrices de $\mathbf{E}_Z^2$, $\mathbf{E}_Y^2$, $\mathbf{E}_X^2$ de (4.3), avec une matrice $\mathbf{E}_Z^2$ conforme au modèle (donc qui vaut l'identité), et une matrice $\mathbf{E}_X^2$ réelle. Ces valeurs des matrices font que les paramètres des mesures ϕ_z^2, ψ_z^2 et ψ_x^2 sont tous les trois conformes aux valeurs cibles de 0, 0 et π respectivement (il reste ϕ_x^2, ϕ_y^2 et ψ_y^2 qui peuvent ne pas être conformes).

référence a 3 paramètres (4 paramètres pour la base $\begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2\end{bmatrix}$ moins 1 pour la phase globale), et on a annulé 3 paramètres de mesures (ϕ_z^1, ψ_z^1 et ψ_y^1) pour la définir, c'est cohérent.

Nous avons donc fait le choix (avec les Conditions 1 et 2) de définir la base de référence à partir des mesures :

- La direction de $|0\rangle$ ou de $\begin{pmatrix}1\\0\end{pmatrix}$ est définie comme la direction du premier vecteur propre de la mesure selon Z.
- La direction de $|1\rangle$ ou de $\begin{pmatrix}0\\1\end{pmatrix}$ est définie comme la direction du deuxième vecteur propre de la mesure selon Z.
- Les phases absolues de $|0\rangle$ et $|1\rangle$ ne sont pas définies, mais on choisit leur phase relative telle que les vecteurs propres de la mesure selon X puissent s'écrire comme des vecteurs réels dans la base de référence.

Comme Z est une mesure projective, on a la garantie que les deux directions sont bien orthogonales, c'est un avantage qui justifie de définir la base de référence avec les mesures et pas avec les états d'entrée.

4.3 Identification des paramètres

4.3.1 Équations

Dans cette section, nous étudions comment estimer les paramètres ϕ_y, ψ_y et ϕ_x de (4.3) en mesurant des copies d'états purs inconnus avec les trois types de mesures X, Y et Z sur un seul qubit. Ces mesures ont les matrices de vecteurs propres de (4.3).

Considérons n_i états purs $\boldsymbol{w}_j = \begin{pmatrix} a_j \\ b_j e^{i\phi_j}\end{pmatrix}$ (a_j et b_j sont des réels positifs, ϕ_j est un angle, $j \leq n_i$ est l'indice de l'état que l'on considère) [3]. Calculons le vecteur qui contient les espérances des fréquences d'occurrence (c.à.d les probabilités théoriques) de 0 et de 1 quand on mesure l'état selon Z, X et Y :

$$\boldsymbol{f}(a_j, \phi_j, \phi_x, \phi_y, \psi_y) = \left| \begin{bmatrix}\mathbf{E}_Z \\ \mathbf{E}_X \\ \mathbf{E}_Y\end{bmatrix} \boldsymbol{w}_j \right|^2$$

en remplaçant les matrices de vecteurs propres par leurs expressions dans (4.3), on a :

$$\boldsymbol{f}(a_j, \phi_j, \phi_x, \phi_y, \psi_y) = \left| \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \cos(\phi_x) & \sin(\phi_x) \\ \sin(\phi_x) & -\cos(\phi_x) \\ \cos(\phi_y) & -\sin(\phi_y)e^{i\psi_y} \\ \sin(\phi_y) & \cos(\phi_y)e^{i\psi_y} \end{pmatrix} \begin{pmatrix} a_j \\ b_j e^{i\phi_j}\end{pmatrix} \right|^2 . \tag{4.5}$$

Le vecteur $\boldsymbol{f}(a_j, \phi_j, \phi_x, \phi_y, \psi_y)$ (que l'on peut estimer avec les mesures) a 5 paramètres (on ne compte pas b_j car il s'exprime en fonction de a_j : $b_j = \sqrt{1-a_j^2}$) et il contient 6 éléments mais

[3] On n'utilise pas la même notation que pour les états de la QPT en section 3.1.1, car les $\boldsymbol{w}_j$ sont des états mono-qubit.

seulement 3 éléments linéairement indépendants (les sommes des deux premiers éléments, des deux éléments du milieu et des deux derniers éléments valent 1). On ne peut donc pas espérer calculer tous les paramètres à partir des mesures sur un seul état.

Si on considère n_i états d'entrée, on a $2n_i + 3$ paramètres (car parmi les 5 paramètres que l'on avait sur un vecteur, 2 dépendaient du vecteur et 3 n'en dépendaient pas) et $3n_i$ mesures indépendantes. Le n_i minimal pour pouvoir espérer estimer tous les paramètres des mesures (et des états mono-qubit mesurés notés $\boldsymbol{w}_1 ... \boldsymbol{w}_{n_x}$, mais ceux-ci ne nous intéressent pas) est $n_i = 3$. Les espérances des fréquences d'occurrences de chaque résultat possible sont contenues dans la matrice :

$$\begin{aligned}
\mathbf{F}(\boldsymbol{a}, \boldsymbol{\phi}, \phi_x, \phi_y, \psi_y) &= \left| \begin{bmatrix} \mathbf{E}_Z \\ \mathbf{E}_X \\ \mathbf{E}_Y \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 & ... & \boldsymbol{w}_{n_x} \end{bmatrix} \right|^2 \\
&= \left| \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \cos(\phi_x) & -\sin(\phi_x) \\ \sin(\phi_x) & \cos(\phi_x) \\ \cos(\phi_y) & -\sin(\phi_y)e^{i\psi_y} \\ \sin(\phi_y) & \cos(\phi_y)e^{i\psi_y} \end{pmatrix} \begin{pmatrix} a_1 & ... & a_{n_i} \\ b_1 e^{i\phi_1} & ... & b_{n_i} e^{i\phi_{n_i}} \end{pmatrix} \right|^2
\end{aligned} \tag{4.6}$$

avec $\boldsymbol{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_{n_i} \end{pmatrix}$ et $\boldsymbol{\phi} = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{n_i} \end{pmatrix}$.

Notre objectif est d'identifier les trois paramètres ϕ_x, ϕ_y, ψ_z (et si nécessaire, les $2n_i$ paramètres des états d'entrée : $a_1, ..., a_{n_x}, \phi_1, ..., \phi_{n_x}$) à partir des espérances des mesures :$\mathbf{F}(\boldsymbol{a}, \boldsymbol{\phi}, \phi_x, \phi_y, \psi_y)$.

Nous n'avons pas réussi à trouver des solutions analytiques (même pour le cas de base $n_i = 3$). Les deux premières lignes de la matrice $\mathbf{F}(\boldsymbol{a}, \boldsymbol{\phi}, \phi_x, \phi_y, \psi_y)$ sont les modules au carré des composantes des états mesurés. L'élément en troisième ligne et j-ième colonne vaut $|a_j \cos(\phi_x) - b_1 e^{i\phi_j} \sin(\phi_x)|^2$ on peut montrer que cette expression se simplifie en :

$$(\mathbf{F}(\boldsymbol{a}, \boldsymbol{\phi}, \phi_x, \phi_y, \psi_y))_{3,j} = \frac{1}{2}\left(1 + c_j \cos(2\phi_x) - d_j \sin(2\phi_x)\cos(\phi_j)\right) \tag{4.7}$$

avec $c_j = a_j^2 - b_j^2$ et $d_j = 2a_j b_j$. La quatrième ligne n'apporte pas plus d'information (les troisième et quatrième lignes somment à 1). On peut montrer que l'élément en 5-ième ligne et j-ième colonne s'exprime comme pour la troisième ligne avec ϕ_x remplacé par ϕ_y, et ϕ_j remplacé par $\phi_j + \psi_y$:

$$(\mathbf{F}(\boldsymbol{a}, \boldsymbol{\phi}, \phi_x, \phi_y, \psi_y))_{5,j} = \frac{1}{2}\left(1 + c_j \cos(2\phi_y) - d_j \sin(2\phi_y)\cos(\phi_j + \psi_y)\right) \tag{4.8}$$

Il est possible d'isoler $\cos(\phi_j)$ avec (4.7) et $\sin(\phi_j)$ avec (4.8) en développant $\cos(\phi_j + \psi_y)$. On peut ensuite éliminer les variables ϕ_j (qui ne nous intéressent pas) en posant $\cos(\phi_j)^2 + \sin(\phi_j)^2 = 1$. Cependant, ces équations sont très complexes et même les logiciels de calcul formel ne nous ont pas permis d'en déduire des expressions de ϕ_x, ϕ_y et ψ_z.

4.3.2 Stratégie de résolution numérique

On pourrait croire que notre incapacité à trouver une solution analytique ou même à savoir pour quelles valeurs des paramètres la solution est unique est un problème, mais ce serait oublier d'où viennent les équations. En pratique, on sait que ϕ_x et ϕ_y sont proches de $\pi/4$, que ψ_y est proche de $\pi/2$, et les valeurs des éléments de $\boldsymbol{a}$ et $\boldsymbol{\phi}$ sont connues et contrôlées (on choisit quels états

préparer) avec une précision raisonnable. On dispose donc d'un point initial raisonnable pour résoudre le problème d'optimisation suivant :

$$\left(\widehat{\phi_x}, \widehat{\phi_y}, \widehat{\psi_y}, \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{\phi}}\right) = \underset{\phi_x,\phi_y,\psi_y,\boldsymbol{a},\boldsymbol{\phi}}{\arg\min} \sum_{j=1}^{n_i} \mathcal{L}^s_{\boldsymbol{f}_k(a_j,\phi_j,\phi_x,\phi_y,\psi_y)}(\mathbf{n}_j) \tag{4.9}$$

où, avec le modèle multinomial, l'opposé de la log-vraisemblance $\mathcal{L}^s_{\boldsymbol{f}_k(a_j,\phi_j,\phi_x,\phi_y,\psi_y)}(\mathbf{n}_j)$ du vecteur $\mathbf{n}_j$ qui contient les résultats (nombre d'occurrences) des mesures sur le j-ième état s'écrit :

$$\mathcal{L}^{exact}_{\boldsymbol{f}_k(a_j,\phi_j,\phi_x,\phi_y,\psi_y)}(\mathbf{n}_j) = -\sum_{k=1}^{n_{proba}} (\boldsymbol{n}_j)_k log\left(\left(\boldsymbol{f}\left(a_j,\phi_j,\phi_x,\phi_y,\psi_y\right)\right)_k\right) \tag{4.10}$$

où $(.)_k$ est le k-ième élément d'un vecteur, $n_{proba} = 3d = 6$ est le nombre de résultats possibles pour les 3 mesures pour un état, $\boldsymbol{f}\left(a_j,\phi_j,\phi_x,\phi_y,\psi_y\right)$ est le vecteur qui contient les probabilités théoriques, défini dans (4.5), et n_c est le nombre de réalisations de chaque type de mesure sur chaque état mesuré. Si on choisit d'utiliser le modèle gaussien régularisé à la place du modèle multinomial :

$$\mathcal{L}^{gauss}_{\boldsymbol{f}_k(a_j,\phi_j,\phi_x,\phi_y,\psi_y)}(\boldsymbol{n}_j) = \sum_{k=1}^{n_{proba}} \frac{\left((\boldsymbol{n}_j)_\ell - n_c\left(\boldsymbol{f}\left(a_j,\phi_j,\phi_x,\phi_y,\psi_y\right)\right)_k\right)^2}{n_c(\widetilde{\boldsymbol{p}}_j)_k} \tag{4.11}$$

où $\widetilde{\boldsymbol{p}}_j = \frac{\boldsymbol{n}_j+5}{n_c+5d}$ contient les probabilités régularisées. Ces deux expressions de la vraisemblance ont été établies dans la section 2.4, réutilisées dans la section 3.3.

Nous verrons dans la section 5.1.6, que, avec un bon point initial, le problème d'optimisation de (4.9) se résout assez facilement avec l'algorithme d'optimisation par descente de gradient (quasi-Newton) que nous avons utilisé pour la maximisation de la vraisemblance dans les Sections 2.4 et 3.3. Nous aurions pu séparer l'optimisation des paramètres des états de l'optimisation principale (comme nous avons fait dans la section 3.3) en résolvant $\left(\widehat{\phi_x}, \widehat{\phi_y}, \widehat{\psi_y}\right) = \underset{\phi_x,\phi_y,\psi_y}{\arg\min} \sum_{j=1}^{n_i} \underset{a_j,\phi_j}{\min} \mathcal{L}^s_{\boldsymbol{f}_k(a_j,\phi_j,\phi_x,\phi_y,\psi_y)}(\mathbf{n}_j)$ plutôt que (4.9) ; mais, étant donné le faible nombre de paramètres : $3 + 2n_i$ (contre $d^2 + 2n_i$ pour l'optimisation de la vraisemblance dans la section 3.3), l'optimisation qui en résulte n'est pas plus rapide ici.

4.3.3 Choix des états mesurés

Dans la présente section nous cherchons à déterminer quels états cibles on peut choisir pour s'assurer que la QMT se passe bien. Ce travail nécessitera de résoudre des problèmes d'optimisation, mais le lecteur doit garder à l'esprit que ces problèmes ne doivent être résolus qu'une seule fois pour choisir les états, et pas à chaque réalisation de la QMT.

Nous n'avons pas l'équivalent de la condition nécessaire et suffisante (3.19) pour la QPT ici, nous n'avons donc a priori aucune idée de comment choisir nos états d'entrée pour être sûr que :

1. la solution du problème d'optimisation (4.9) soit unique.

2. L'algorithme de descente du gradient que nous utilisons converge vers le vrai minimum même si l'on part d'un point qui est loin du minimum.

3. Le problème soit bien conditionné (i.e. que l'on ne puisse pas avoir une petite erreur sur les mesures qui génère de grosses erreurs sur les paramètres estimés).

Parmi ces 3 critères, le premier est le plus difficile à vérifier (il faudrait réaliser une optimisation globale sur un critère non convexe pour vérifier qu'il n'y a pas de meilleure solution) et n'est pas très important. Il était important pour la QPT dans la section 3.2.1 parce que l'on n'avait pas d'a priori sur la valeur du processus à identifier. Mais, ici, nous connaissons à peu près les paramètres des mesures X Y et Z, et nous voulons juste affiner en estimant leurs vraies valeurs des paramètres. Le fait d'avoir plusieurs solutions n'est pas gênant si elles ne sont pas trop proches (il faut juste que l'on tombe sur la bonne solution à partir du point initial), et si le Critère 2 est respecté, alors nous saurons qu'il n'existe pas d'autre solution au voisinage du point initial.

Le Critère 2 peut être vérifié en réalisant plusieurs optimisations avec des points initiaux autour des vrais paramètres (connus en simulation).

Le Critère 3 est très facile (d'un point de vue calculatoire) à vérifier, en effet, nous avons expliqué dans la section 3.3.5 comment la borne de Cramér-Rao peut être calculée. Ici, le critère que l'on minimise est un peu différent (pas de processus dont les paramètres sont à identifier, et les matrices de vecteur propres ne sont pas fixées), mais l'idée est la même : on échantillonne les réalisations d'une loi multinomiale dont les probabilités théoriques sont fonctions des paramètres à identifier. On adapte (3.41) la formule de la borne de l'information de Fisher (dont la borne de Cramér-Rao est l'inverse) de la loi régularisée au problème de QMT :

$$\mathbf{I}_{QMT}^{Gauss}(\boldsymbol{\theta}) = n_c \mathbf{J}_{QMT}(\boldsymbol{\theta})^* \begin{pmatrix} \widetilde{p}_1(\boldsymbol{\theta}) & & \\ & \ddots & \\ & & \widetilde{p}_{n_{prob}}(\boldsymbol{\theta}) \end{pmatrix}^{-1} \mathbf{J}_{QMT}(\boldsymbol{\theta}) \tag{4.12}$$

où n_{prob} est le nombre de probabilités mesurées, ici $n_{prob} = 6n_i$, les $\widetilde{p}_k$ sont les probabilités régularisées qui sont fonctions des vrais paramètres $\boldsymbol{\theta}$, ici (par opposition à (3.41)) $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{\phi} \\ \phi_x \\ \phi_y \\ \psi_y \end{pmatrix}$ a $2n_i + 3$ éléments, (nous l'appelons $\boldsymbol{\theta}$ et non $\boldsymbol{\theta}_0$ car nous allons choisir les éléments de $\boldsymbol{a}$ et $\boldsymbol{\phi}$) et $\mathbf{J}_{QMT}(\boldsymbol{\theta})$ est la jacobienne de taille $n_{prob} \times (2n_i + 3)$ qui contient les dérivées de toutes les probabilités théoriques en fonction des $2n_i + 3$ paramètres. Nous avons choisi d'utiliser le modèle gaussien régularisé et non le modèle multinomial, car ce dernier modélise la variance de l'erreur des mesures qui ont une probabilité associée de 1 ou 0 comme nulle. Avec le modèle multinomial, un algorithme d'optimisation qui minimise la variance des estimées de QMT aura donc tendance à sélectionner des états de la base de mesure pour lesquels les résultats de mesure sont déterministes même si le conditionnement est très mauvais (en pratique, il l'est, car les mesures dépendent très peu des états) car le mauvais conditionnement devrait être compensé par le fait que la variance de l'erreur de mesure est nulle ; et c'est le cas pour le modèle multinomial, mais en pratique, avec des erreurs de modélisation (les mesures ne sont pas exactement des mesures projectives, les états ne sont pas exactement purs), on a toujours des erreurs résiduelles. Ces erreurs résiduelles sont modélisées dans le modèle gaussien régularisé, car la régularisation qu'il effectue a pour effet d'augmenter les petites probabilités.

À partir de l'information de Fisher, on peut calculer la borne de Cramér-Rao. L'évaluation de la borne de Cramér-Rao pour un $\boldsymbol{\theta}$ donné est très rapide ($\sim 3ms$ sur un processeur Intel i7-8650U $1,90$GHz). Le critère (scalaire) que nous choisissons pour caractériser la qualité de l'estimation des paramètres est la somme des trois derniers éléments de la diagonale de la borne de Cramér-Rao : c'est la somme des variances des erreurs sur les trois paramètres qui nous intéressent. Nous appelons ce critère $c(\boldsymbol{\theta})$.

Nous choisissons de sélectionner les paramètres $\boldsymbol{a}$ et $\boldsymbol{\phi}$ des états (cibles) mesurés de façon à minimiser $c(\boldsymbol{\theta})$. Bien entendu, comme $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{\phi} \\ \phi_x \\ \phi_y \\ \psi_y \end{pmatrix}$, la quantité à minimiser $c(\boldsymbol{\theta})$ dépend aussi des valeurs de ϕ_x, ϕ_y, ψ_y. Nous considérons que ces paramètres valent les valeurs attendues ($\phi_x = \pi/4$, $\phi_y = \pi/4$ et $\psi_y = \pi$) [4]. Formellement :

$$\begin{pmatrix} \boldsymbol{a}_0 \\ \boldsymbol{\phi}_0 \end{pmatrix} = \underset{\boldsymbol{a},\boldsymbol{\phi}}{\arg\min} \; c \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{\phi} \\ \pi/4 \\ \pi/4 \\ \pi \end{pmatrix}. \tag{4.13}$$

Ce problème d'optimisation est résolu avec le même algorithme de descente de gradient (quasi-Newton BFGS) que l'optimisation de la vraisemblance, sauf que les gradients sont estimés par la méthode des différences finies (ils étaient calculés analytiquement pour la vraisemblance). Nous effectuons le changement de variable $a \leftarrow \tan\left(\pi(a - \frac{1}{2})\right)$ (voir section 2.4 pour une explication détaillée) pour ne pas avoir de contrainte sur les paramètres à optimiser. Étant donné le faible nombre de paramètres à optimiser ($2n_i$), la minimisation est très rapide ($\sim 0.4s$ pour $n_i = 3$), et il est réaliste de tenter de réaliser une optimisation globale en lançant quelques centaines d'optimisations sur des paramètres aléatoires.

Pour $n_i = 3$, nous lançons 1000 optimisations à des points initiaux aléatoires qui trouvent chacune un minimum local. Sur les 1000 valeurs minimales, 779 sont identiques (les autres sont supérieures). Ces 779 valeurs identiques sont atteintes pour 170 valeurs des paramètres distinctes (les sauts de 2π dans les paramètres de phases ne sont pas considérés comme distincts). Si l'on enlève les valeurs des paramètres qui sont les mêmes à des permutations sur les états près (par exemple en échangeant le premier état et le second), il reste 32. On affiche toutes les valeurs que peuvent prendre les 3 états en figure 4.1 sur la sphère de Bloch. La sphère de Bloch est une représentation en trois dimensions des états quantiques à un qubit. Les 3 axes de référence de l'espace de la sphère de Bloch sont les trois premiers vecteurs propres des mesures X, Y et Z, et si les angles $\psi_b \in [0, \pi]$ et $\theta_b \in [0, 2\pi]$ sont tels que les coordonnées d'un point sur la sphère de Bloch sont $\begin{pmatrix} \sin(\psi_b)\cos(\theta_b) \\ \sin(\psi_b)\sin(\theta_b) \\ \cos(\psi_b) \end{pmatrix}$, alors l'état quantique associé est $\cos(\psi_b/2)\,|0\rangle + e^{i\theta_b}\sin(\psi_b/2)\,|1\rangle$.

En pratique les 32 combinaisons d'états peuvent toutes être déduites (nous allons voir en quel sens) d'une seule solution, les 3 états quantiques associés à cette solution sont (arrondis) :

$$\boldsymbol{w}_1^3 = \begin{pmatrix} 0.8883 \\ 0.3247 + 0.3247i \end{pmatrix} \boldsymbol{w}_2^3 = \begin{pmatrix} 0.8069 \\ -0.2324 + 0.5430i \end{pmatrix} \boldsymbol{w}_3^3 = \begin{pmatrix} 0.8069 \\ 0.5430 - 0.2324i \end{pmatrix}. \tag{4.14}$$

Les 3 transformations suivantes peuvent être appliquées sur les 3 états de (4.14) pour retrouver

[4] En pratique, ce ne sera pas exactement le cas, et les états mesurés ne seront pas exactement les états associés aux paramètres $\boldsymbol{a}$ et $\boldsymbol{\phi}$ que nous allons choisir, mais on devrait être assez proche pour que $c(\boldsymbol{\theta})$ soit faible.

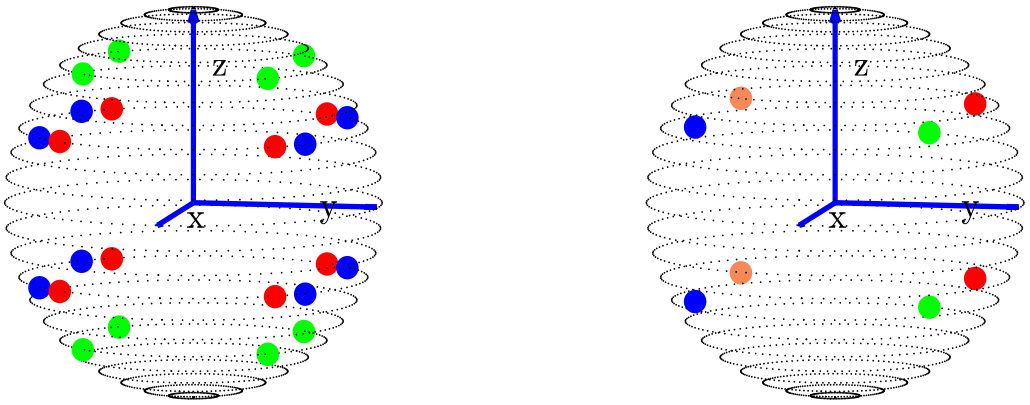


FIGURE 4.1 : Valeurs prises par les états qui minimisent l'erreur d'estimation de la QMT (une solution est composée d'un point rouge un point vert et un point bleu à gauche et de points rouge verts bleus et oranges à droite, ces points représentent les 3 et 4 états qui sont les solutions de (4.13) pour 3 et 4 états) sur la sphère de Bloch avec $n_i = 3$ (à gauche) et $n_i = 4$ (à droite) états d'entrée. Si un point correspond à une valeur qui a été prise plusieurs fois (en pratique, c'est le cas pour tous les points représentés), il garde la dernière couleur (sur la figure de droite par exemple, il y a 16 points de chaque couleur, mais comme ils prennent souvent les mêmes valeurs, on n'en voit que 2). Les vecteurs $\boldsymbol{x}$, $\boldsymbol{y}$ et $\boldsymbol{z}$ sont les états qui donnent 0 (ceux qui donnent 1 sont situés à l'opposé sur la sphère de Bloch) avec probabilité 1 quand on les mesure selon X, Y et Z respectivement, c'est-à-dire $|0\rangle$, $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ et $\frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$.

les 32 états solutions de (4.13) :

$$\begin{aligned} \boldsymbol{f}_x : \begin{pmatrix} a_1 \\ a_2 + b_2 i \end{pmatrix} &\rightarrow \begin{pmatrix} a_1 \\ -a_2 + b_2 i \end{pmatrix} \\ \boldsymbol{f}_y : \begin{pmatrix} a_1 \\ a_2 + b_2 i \end{pmatrix} &\rightarrow \begin{pmatrix} a_1 \\ a_2 - b_2 i \end{pmatrix} \\ \boldsymbol{f}_z : \begin{pmatrix} a_1 \\ a_2 + b_2 i \end{pmatrix} &\rightarrow \frac{a_2 + b_2 i}{|a_2 - b_2 i|} \begin{pmatrix} a_2 - b_2 i \\ a_1 \end{pmatrix}. \end{aligned} \tag{4.15}$$

Les fonctions sont appelées ainsi parce qu'elles sont les symétries par rapport aux plans passant par l'origine et de vecteurs normaux $\boldsymbol{x}$, $\boldsymbol{y}$ et $\boldsymbol{z}$ respectivement sur la sphère de Bloch. Du point de vue des types de mesures X, Y et Z, mesurer l'état représenté par un vecteur $\boldsymbol{v}$ selon X (resp. Y et Y) donne les mêmes valeurs de probabilités, mais inversées (la probabilité d'avoir 0 devient la probabilité d'avoir 1) que quand on mesure $\boldsymbol{f}_x(\boldsymbol{x})$ (resp. $\boldsymbol{f}_y(\boldsymbol{y})$ et $\boldsymbol{f}_z(\boldsymbol{z})$).

Pour $n_i = 4$, les résultats sont plus faciles à analyser, (4.13) n'a que 16 solutions uniques (trouvées aussi avec 1000 optimisations), elles sont représentées sur la figure 4.1 à droite. De nombreux points sont superposés, on ne voit que les quatre points de la première solution (en haut) et la dernière (en bas). Les valeurs des états correspondant à la première solution sont :

$$\boldsymbol{w}_1^4 = \begin{pmatrix} 0.8510 \\ 0.3714(1+i) \end{pmatrix} \boldsymbol{w}_2^4 = \begin{pmatrix} 0.8510 \\ 0.3714(1-i) \end{pmatrix} \boldsymbol{w}_3^4 = \begin{pmatrix} 0.8510 \\ -0.3714(1-i) \end{pmatrix} \boldsymbol{w}_4^4 = \begin{pmatrix} 0.8510 \\ -0.3714(1+i) \end{pmatrix}. \tag{4.16}$$

On choisit de considérer les états de (4.14) et (4.16) parmi les solutions (minimas locaux) de (4.9) pour $n_i = 3$ et $n_i = 4$ que nous avons trouvés. C'est un choix arbitraire. Idéalement, nous

aurions voulu des états qui permettent aussi de garantir que l'optimisation de (4.9) ne donne pas un autre minimum que celui qui est voisin des vraies valeurs des paramètres si on se trompe sur l'initialisation. Afin de savoir si c'est le cas, étudions comment l'algorithme d'optimisation que nous avons choisi pour résoudre (4.9) avec $n_i = 3$ et $n_i = 4$ converge avec les états cibles de (4.14) et (4.16) respectivement si on introduit des erreurs raisonnables et que l'on initialise l'algorithme de descente de gradient à un mauvais point. Pour ce faire, on simule le système où les états cibles sont ceux de (4.14) (resp. (4.16)) mais l'opérateur les prépare de façon imparfaite, et les vrais états préparés sont les états cibles multipliés à gauche par une matrice unitaire aléatoire $\begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r)e^{i\phi_r^1} \\ \sin(\theta_r)e^{i\phi_r^2} & \cos(\theta_r)e^{i(\phi_r^1+\phi_r^2)} \end{pmatrix}$ où les angles θ_r ϕ_r^1 et ϕ_r^2 sont modélisés comme des variables aléatoires gaussiennes indépendantes centrées et d'écart types σ_{prep} que nous allons faire varier. On simule aussi des types de mesures quantiques X, Y et Z qui devraient être conformes au modèle, mais à la place, on a $\mathbf{E}_Z^2 = \mathbf{E}(0,0), \mathbf{E}_Y^2 = \mathbf{E}(\phi_y, \psi_y), \mathbf{E}_X^2 = \mathbf{E}(\phi_x, \pi)$ où les angles ϕ_y ψ_y et ϕ_x sont modélisés comme des variables aléatoires gaussiennes indépendantes centrées en les valeurs cibles, c'est-à-dire $\pi/4$, $\pi/2$ et $\pi/4$ respectivement, et d'écart type σ_{mes}. En simulation, nous choisissons de toujours considérer $\sigma_{mes} = \sigma_{prep}$ (arbitraire) mais nous réalisons des simulations avec 100 valeurs différentes de σ_{mes} (et σ_{prep}) allant de 0 à 0,5. Pour chaque valeur de σ_{mes}, nous réalisons 1000 simulations avec des réalisations différentes des variables aléatoires qui déterminent les vrais états d'entrée et les vrais paramètres des mesures, nous simulons aussi un nombre fini de mesures avec $n_c = 1000$ (n_c et le nombre de copies de chaque valeur d'état mesurée par type de mesure) si $n_i = 3$ et $n_c = 750$ si $n_i = 4$ (pour que le nombre total d'états à préparer $n_i n_c$ soit le même). Les résultats des mesures sont modélisés comme des variables aléatoires multinomiales. Nous estimons les paramètres des types de mesures ϕ_y ψ_y et ϕ_x en résolvant (4.9) avec la version gaussienne de la vraisemblance. L'algorithme d'optimisation par descente de gradient est d'abord initialisé aux valeurs cibles des paramètres (vrais paramètres des états d'entrée et $\phi_y = \pi/4$, $\psi_y = \pi/2$ $\phi_x = \pi/4$) pour obtenir les estimées $\widehat{\phi_{y_{ML}}}, \widehat{\psi_{y_{ML}}}, \widehat{\phi_{x_{ML}}}$, puis initialisé aux vraies valeurs des paramètres (seulement connus en simulation) pour obtenir $\widehat{\phi_{y_{ref}}}, \widehat{\psi_{y_{ref}}}, \widehat{\phi_{x_{ref}}}$. Nous considérons que la résolution de (4.9) avec le point initial des valeurs cible a convergé vers le bon minimum si $\widehat{\phi_{y_{ML}}} = \widehat{\phi_{y_{ref}}}, \widehat{\psi_{y_{ref}}} = \widehat{\psi_{y_{ref}}}$ et $\widehat{\phi_{x_{ML}}} = \widehat{\phi_{x_{ref}}}$.

La figure 4.2 représente la probabilité que la résolution de (4.9) converge vers le bon minimum avec le point initial des valeurs cibles en fonction de σ_{mes}. Cette probabilité est notée p_{conv}. On la représente pour les $n_i = 3$ et $n_i = 4$ états d'entrée de (4.14) et (4.16) respectivement.

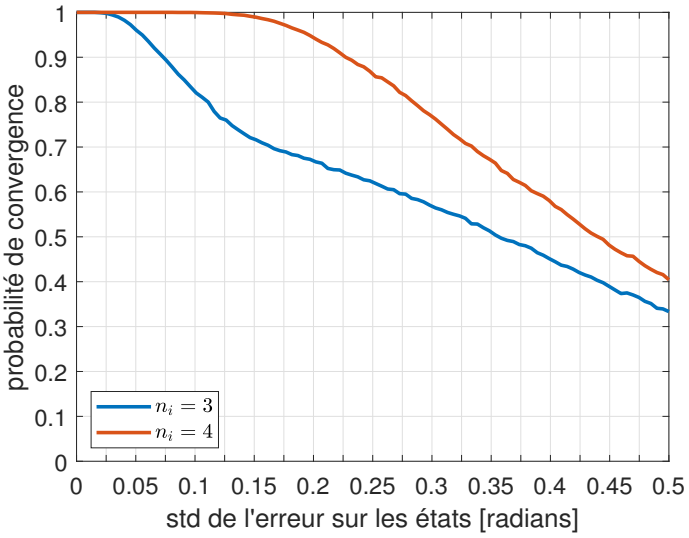


FIGURE 4.2 : Probabilité que l'optimisation de (4.9) converge vers le bon minimum

p_{conv} reste égale à 1 (c.à.d. l'algorithme de maximum de vraisemblance converge toujours vers le bon minimum) tant que σ_{mes} est inférieur à $0,12$ pour $n_i = 4$ et $0,025$ pour $n_i = 3$. Si on veut garantir que l'algorithme continue de bien converger avec un σ_{mes} plus grand, on peut considérer plus d'états mesurés ($n_i = 5$ par exemple) ou alors considérer des états différents de (4.14) et (4.16). Cependant, nous considérons que les performances sont satisfaisantes avec les $n_i = 4$ états d'entrée de (4.16).

4.4 Plus d'un qubit et lien avec la QPT

On considère un système multi-qubit pour lequel le problème d'optimisation de (4.9) a été résolu pour chaque qubit qui compose le système. Nous avons donc $3n_{qb}$ matrices de vecteurs propres 2×2 : $\mathbf{E}_{X_1}, \mathbf{E}_{Y_1}, \mathbf{E}_{Z_1}, ..., \mathbf{E}_{X_{n_{qb}}}, \mathbf{E}_{Y_{n_{qb}}}, \mathbf{E}_{Z_{n_{qb}}}$. Nous supposons que toutes les mesures mono-qubit sont bien non-intriquées, cette hypothèse était faite de façon implicite dans les parties précédentes, car nous considérions que les états mesurés se comportaient comme des états purs sur un qubit.

Pour avoir les matrices de vecteurs propres des mesures multi-qubit on peut donc calculer le produit tensoriel des matrices de vecteurs propres mono-qubit. Par exemple :

$$\begin{aligned} \mathbf{E}_{X...X} &= \mathbf{E}_{X_1} \otimes \mathbf{E}_{X_2} \otimes ... \otimes \mathbf{E}_{X_{n_{qb}}} \\ \mathbf{E}_{Y....Y} &= \mathbf{E}_{Y_1} \otimes \mathbf{E}_{Y_2} \otimes ... \otimes \mathbf{E}_{Y_{n_{qb}}} \\ \mathbf{E}_{Z....Z} &= \mathbf{E}_{X_1} \otimes \mathbf{E}_{Z_2} \otimes ... \otimes \mathbf{E}_{Z_{n_{qb}}}. \end{aligned} \tag{4.17}$$

On peut réaliser la QMT avant de faire la QPT sur les mesures mono-qubit du dispositif de QPT en se servant des valeurs des paramètres des portes estimés dans l'algorithme du maximum de vraisemblance de la QPT (et de la QST de chaque état).

Mais on peut aussi réaliser la QMT et la QPT avec le même dispositif. Par exemple, pour deux qubits, on peut utiliser le dispositif de la figure 4.3 :

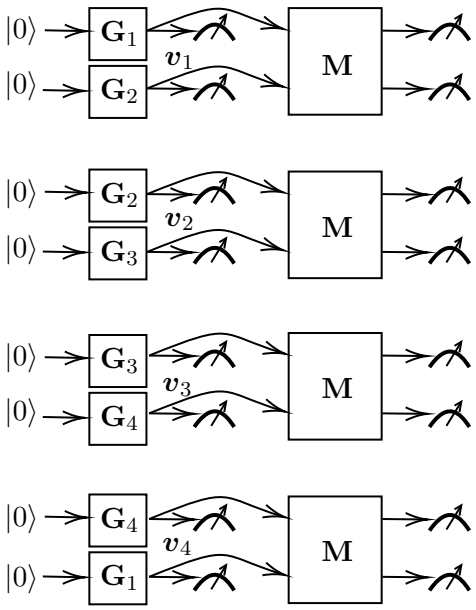


FIGURE 4.3 : Circuit quantique à réaliser pour pouvoir réaliser la QPT et la QMT simultanément. L'extension pour des processus à plus de 2 qubits est simple (on crée les n_i états à partir des 4 portes), il faut juste vérifier que les états d'entrée satisfont (3.19) pour que la QPT soit possible.

Les portes quantiques mono-qubit représentées par les matrices $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4$ dans la figure 4.3 sont telles que les états mono-qubit préparés soient ceux de (4.16) : $\mathbf{G}_j \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \boldsymbol{w}_j^4 \ \forall j \in 1, .., 4.$ Les quatre états initiaux à deux qubits avec lesquels nous faisons la QPT sont : $\boldsymbol{v}_1 = \boldsymbol{w}_1^4 \otimes \boldsymbol{w}_2^4$, $\boldsymbol{v}_2 = \boldsymbol{w}_2^4 \otimes \boldsymbol{w}_3^4$, $\boldsymbol{v}_3 = \boldsymbol{w}_3^4 \otimes \boldsymbol{w}_4^4$, $\boldsymbol{v}_4 = \boldsymbol{w}_4^4 \otimes \boldsymbol{w}_1^4$. Il se trouve que ces états vérifient la condition (3.20), ils forment une base et tous les états sont non-orthogonaux deux à deux, la QPT est donc possible. Pour réaliser la QMT, il faut que les états mono-qubit $\boldsymbol{w}_1^4, \boldsymbol{w}_2^4, \boldsymbol{w}_3^4, \boldsymbol{w}_4^4$ soient mesurés avec les types de mesures à identifier X, Y, et Z sur le premier et sur le deuxième qubit (car on considère que la mesure Y (par exemple) n'a pas les mêmes paramètres sur le premier et sur le deuxième qubit). Cette condition n'est pas respectée si on choisit d'utiliser les $2n_{qb} + 1 = 5$ types de mesures de la section 2.3 sur les états à 2 qubits, c'est-à-dire ZZ, ZX, ZY, XX, XY ; en effet, avec ces mesures, le premier qubit n'est jamais mesuré selon Y. Cependant, si l'on effectue les 4 types de mesures de la section 2.2, c'est-à-dire XX, YY, ZZ, XY pour deux qubits, alors, il est possible de réaliser la QMT avec la QPT. En pratique, les paramètres de tous les types de mesures sont estimés ce qui nous permet d'avoir une nouvelle estimée de la matrice $\mathbf{A}$ (qui est la concaténation des matrices de vecteurs propres des mesures effectuées, voir (2.4) et (2.11)). Il se trouve que tous les algorithmes utilisés ensuite (de phaseCut pour la QST au maximum de vraisemblance pour la QPT) peuvent fonctionner avec toute matrice $\mathbf{A}$ tant qu'elle est connue (il faut aussi que $\boldsymbol{v} \to |\mathbf{A}\boldsymbol{v}|^2$ soit injective à une phase globale près, mais nous avons vu dans la section 2.2.2 que si $\mathbf{A}$ a assez de lignes, c'est presque toujours le cas), on peut donc faire marcher tous ces algorithmes avec la version de $\mathbf{A}$ corrigée par la QPT.

4.5 Conclusion

Dans ce chapitre, nous avons étudié comment les paramètres de nos mesures mono-qubit peuvent être estimés en mesurant des états que l'on ne connaît que de façon imprécise avec nos mesures (aussi connues de façon imprécise). Nous avons montré que, si on se permet de définir la base de référence en fonction des mesures X, Y et Z pour chaque qubit, alors, pour chaque qubit, on peut estimer tous les paramètres des mesures en mesurant au moins 3 états. Nous avons étudié comment les états d'entrée peuvent être choisis pour maximiser la précision de l'estimation des paramètres des mesures, et nous avons choisi les 4 états de (4.16) sur chaque qubit. Finalement, nous avons vu que, pour étendre notre algorithme de QMT mono-qubit à des mesures non intriquées multi-qubit, on peut considérer la configuration de la figure 4.3 qui permet de réaliser la QPT et la QMT simultanément. Comme nous privilégions cette configuration, nous ne testerons notre algorithme de QMT que avec la QPT dans le chapitre 5.

Chapitre 5

Validations des algorithmes de tomographie de processus

Sommaire

Les algorithmes de QST qui servent de brique de base pour la QPT ont été testés dans la section 2.5. Les conclusions auxquelles nous sommes arrivés en fin de cette section sont les suivantes :

- Les configurations avec les 4 types de mesures de la section 2.2 et avec les $2n_{qb}+1$ types de mesures de la section 2.3 sont comparables. Mais nous préférons la seconde configuration, nous l'utilisons par défaut.
- Minimiser $\mathcal{L}^{gauss}$ (la version gaussienne régularisée de la vraisemblance) n'a que très peu d'inconvénients (légère perte de précision si les mesures ne sont répétées que peu de fois par rapport à la vraie vraisemblance), et a l'avantage d'être plus robuste, et plus adaptée à la présence d'imperfections non-modélisées.

Dans le présent chapitre, nous évaluons les perfomances de nos algorithmes de QPT (et, dans une moindre mesure, notre algorithme qui réalise la QMT et la QPT simultanément). La configuration de base vers laquelle nous allons revenir le plus souvent est celle de la figure 3.4 avec $n_{qb} = 2$. Cela signifie que l'on prépare les 4 états initiaux dont les valeurs cibles sont celles de (3.21), que nous réecrivons ici :

$$\boldsymbol{v}_1^{tg} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{v}_2^{tg} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{v}_3^{tg} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \boldsymbol{v}_4^{tg} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Nous mettons cette configuration en avant parce que la QPT à deux qubits est un problème classique et, comme nous l'avons vu dans la section 3.2.3, les quatre états initiaux (3.21) permettent d'identifier tous les processus unitaires et sont particulièrement simples à réaliser physiquement.

Nous estimons les paramètres du processus avec la combinaison de l'algorithme de la section 3.1.2, qui donne $\widehat{\mathbf{M}}_{LS}$ puis l'algorithme de maximum de vraisemblance gaussienne de la section 3.3, qui utilise $\widehat{\mathbf{M}}_{LS}$ et donne $\widehat{\mathbf{M}}_{ML}$. Nous souhaitons tester la résistance à l'erreur multinomiale (erreur générée par le nombre de mesures) dans la section 5.1.1. Ensuite, dans la section 5.1.2, nous testons notre résistance aux erreurs systématiques sur les états d'entrée. Nous comparons également notre algorithme à celui de Baldwin et al. [BKD14] dans la section 5.1.3. Dans les sections 5.1.4 et 5.1.5, nous testons la résistance de notre algorithme à des erreurs qui ne sont pas prises en compte par notre modèle de mesure (états intriqués, états mélange, processus non-unitaire). Nous testons notre algorithme de QMT en section 5.1.6 (seule section où la QMT est testée). Pour finir avec la configuration avec $n_{qb} = 2$ qubits et $n_i = 4$ états initiaux, nous testerons la validité de la borne de Cramér-Rao en section 5.1.7 que nous pouvons calculer en sortie de QPT.

Dans la section 5.1.8, nous sortons du cadre de la figure 3.4 et des états (3.21). En effet, nous étendons la configuration testée en augmentant le nombre de qubits, et testons aussi la configuration de la figure 3.5 avec un seul état initial.

Finalement, en section 5.2, nous revenons sur la configuration de la figure 3.4, et testons les algorithmes de QPT avec des données réelles.

5.1 Performances en simulation

5.1.1 Impact du nombre de copies

Nous avons conçu des simulations pour déterminer le nombre (n_c) de copies par type de mesure sur chaque valeur d'état mesuré, dont nous avons besoin dans le cas d'utilisation de la figure 3.5 afin d'obtenir une bonne estimation de la matrice unitaire associée à la porte quantique. Nous commençons par considérer que le nombre fini de mesures est la seule source d'erreurs. Il n'y a pas de décohérence, les mesures suivent le modèle, et les états initiaux que nous considérons sont préparés avec des portes de Hadamard parfaites (comme nous le verrons dans la section 5.1.2, ce dernier point n'est pas vraiment important). Nous faisons varier n_c de 20 à 25 000, et pour chaque valeur de n_c, nous générons 5000 portes quantiques aléatoires. Chacune de ces portes est définie par une matrice unitaire créée en appliquant l'algorithme de Gram-Schmidt à une matrice complexe aléatoire dont chaque coefficient est i.i.d. suivant la distribution normale complexe centrée circulairement symétrique. Pour chaque n_c, nous générons également 5000 (un par porte) ensembles de $n_s n_i n_t$ (un pour chaque valeur unique d'état mesuré et par type de mesure) comptes de mesures $\mathbf{N}$ aléatoires associés à chaque n_c suivant la distribution multinomiale à d résultats possibles. Les espérances de chaque résultat sont fixées à leurs valeurs théoriques (connues uniquement en simulation). Par exemple, si l'un des états mesurés est représenté par $\boldsymbol{v}_{j,k} = \begin{pmatrix} 0.5 & 0.5 & 0.5i & 0.5i \end{pmatrix}^T$ (à identifier avec les mesures) et que nous le mesurons dans la base de calcul (type de mesure ZZ dans la section 2.1.1) avec $n_c = 50$ copies de cet état mesuré dans cette base, nous simulons alors les comptes de mesures en échantillonnant une distribution multinomiale avec des paramètres de probabilités $\boldsymbol{p} = |\mathbf{I}_4 \boldsymbol{v}_{j,k}|^2 = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}^T$ pour les 4 résultats et $n_c = 50$ lancers, les nombres de mesures empiriques qui en résultent pourraient être $\begin{pmatrix} 13 & 10 & 15 & 12 \end{pmatrix}^T$ par exemple.

Pour chaque n_c, nous avons donc 5000 matrices unitaires $\{\mathbf{M}\}$ à estimer et 5000 ensembles

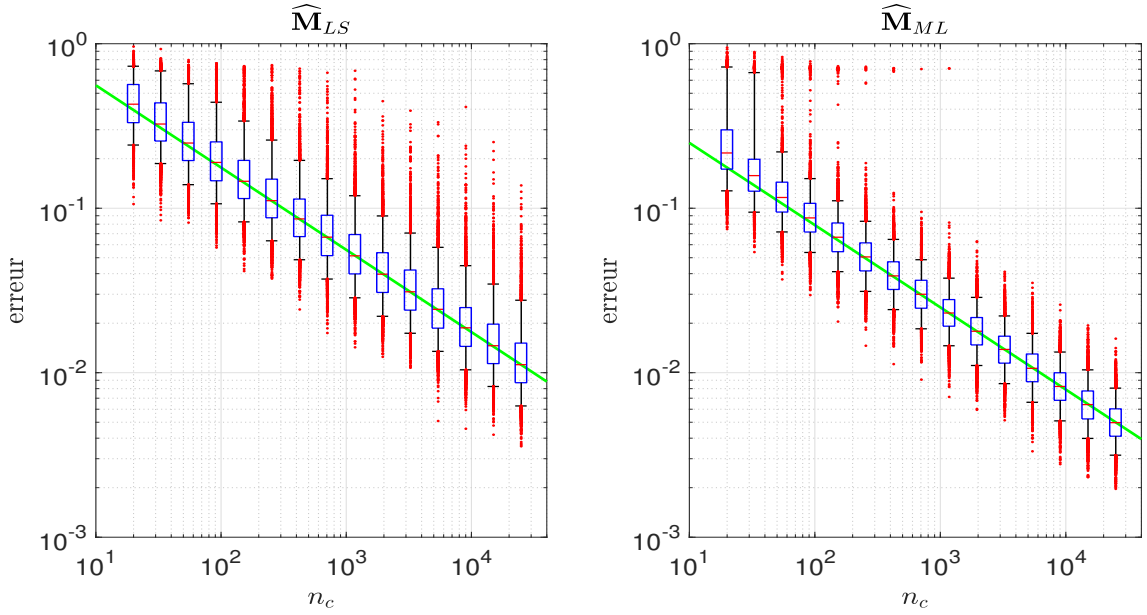


FIGURE 5.1 : Boîtes à moustaches de l'erreur de QPT lorsque le nombre de mesures par état et par type de mesure n_c augmente. La ligne verte (ligne claire du haut à gauche au bas à droite) représente la fonction $n \longrightarrow \frac{c}{\sqrt{n}}$ où c est calculé de manière à ce que la ligne corresponde (au sens des moindres carrés) à la médiane dans la boîte à moustache associée à $n_c \geq 1000$. Le rectangle central (bleu) représente l'espace entre le premier et le dernier quartile, où se trouvent la moitié des observations. La petite ligne (rouge) au milieu de chaque rectangle est la médiane, et les moustaches (noires) vont du premier au dernier 5 centiles. Par définition, 90% des échantillons se situent dans l'intervalle des moustaches. Les performances de l'algorithme d'initialisation (qui donne $\widehat{\mathbf{M}}_{LS}$) sont sur le graphe de gauche, et celles de l'algorithme du maximum de vraisemblance (qui donne $\widehat{\mathbf{M}}_{ML}$) sont sur le graphe de droite.

de mesures sur lesquels effectuer la QPT. Nous exécutons notre algorithme et obtenons 5000 estimations $\{\widehat{\mathbf{M}}_{LS}\}$ et $\{\widehat{\mathbf{M}}_{ML}\}$. Nous devons ensuite quantifier l'erreur entre les $\{\mathbf{M}\}$ et les $\{\widehat{\mathbf{M}}\}$ (avec la convention $\widehat{\mathbf{M}}$ vaut $\widehat{\mathbf{M}}_{LS}$ ou $\widehat{\mathbf{M}}_{ML}$). Nous choisissons d'utiliser la métrique suivante, notée $\mu_p(\widehat{\mathbf{M}}_{LS}, \mathbf{M})$ et appelée erreur :

$$\mu_p(\widehat{\mathbf{M}}_{LS}, \mathbf{M}) = \frac{1}{\sqrt{2d}} ||\mathbf{M} - \widehat{\mathbf{M}}_{LS} e^{i\phi}|| \tag{5.1}$$

où ϕ est l'angle qui minimise l'erreur (il tient compte du fait que $\mathbf{M}$ ne peut être récupéré qu'à une phase globale près) et $||.||$ est la norme de Frobenius. Cette métrique se situe entre 0 (si $\widehat{\mathbf{M}}$ et $\mathbf{M}_{true}$ sont identiques à une phase globale près) et 1 (s'ils sont orthogonaux par rapport au produit scalaire de Hilbert-Schmidt). On peut montrer que $\phi = arg\big(tr(\widehat{\mathbf{M}}_{LS}^H \mathbf{M})\big)$ (arg est la phase d'un nombre complexe et tr est la trace).

Dans la littérature, la fidélité $f(\widehat{\mathbf{M}}, \mathbf{M})$ est plus souvent utilisée, voir par exemple l'équation [BKD14] (24), définie pour tous les processus quantiques (et pas seulement les processus unitaires). La fidélité d'un processus $\widehat{\epsilon}$ par rapport à un processus ϵ est définie comme la fidélité de l'état quantique (voir section 1.1.9) $\boldsymbol{\rho}_{\widehat{\epsilon}}$ (associé à $\widehat{\epsilon}$ par l'isomorphisme de Choi-Jamiolkowski (1.13) par rapport à l'état $\boldsymbol{\rho}_{\epsilon}$ (associé à ϵ). On montre dans l'Annexe C.1 que les deux métriques sont liées lorsqu'il s'agit de processus unitaires $f(\widehat{\mathbf{M}}, \mathbf{M}) = (1 - \mu_p(\widehat{\mathbf{M}}, \mathbf{M})^2)$. Nous préférons μ_p à f parce que μ_p a une signification tangible (un μ_p de $0,1$ peut être vu comme une erreur de 10%). Il est également beaucoup plus instructif lorsque l'estimation $\widehat{\mathbf{M}}$ commence à se rap-

procher de $\mathbf{M}$, car une erreur μ_p de $0,1$ est bien pire qu'une erreur 0,01 (la norme de l'erreur rephasée est dix fois plus grande), mais il est plus difficile de comparer les f associés de 0,99 et 0,9999 (on pourrait penser que ces deux valeurs signifient que l'estimation est très bonne, mais $f = 0{,}99$ correspond à une grosse erreur).

La figure 5.1 montre les boîtes à moustaches (qui affichent la médiane, les premiers et derniers quartiles, les premiers et derniers 5 centile et valeurs qui les dépassent) des 5000 échantillons de l'erreur pour chaque valeur de n_c, pour l'estimateur initial ($\widehat{\mathbf{M}}_{LS}$) et l'estimateur du maximum de vraisemblance ($\widehat{\mathbf{M}}_{ML}$).

Avec l'échelle logarithmique, on retrouve la relation inverse classique entre l'erreur d'estimation et la racine carrée du nombre d'échantillons (la droite qui représente le résultat d'une régression linéaire entre les deux s'ajuste très bien aux dernières médianes) si n_c est suffisamment élevé ($n_c \geq 100$). Le gain apporté par l'algorithme du maximum de vraisemblance apparaît clairement sur la figure 5.1, en pratique, on gagne un facteur supérieur à 2 (erreur réduite de $\sim 55\%$).

5.1.2 Impact de l'erreur systématique

Nous réalisons ici une seconde simulation afin de voir ce qui se passe si nous ajoutons une erreur systématique aux états initiaux. Nous simulons les états initiaux de (3.21), nous fixons $n_c = 1000$, et nous simulons une erreur systématique, chacune des $n_t n_c n_s$ copies correspondant à un état donné a la même erreur systématique. Physiquement, cela signifie que l'initialisation des états à $|0\rangle$ ou les portes de Hadamard utilisées pour les transformer ont des erreurs, mais font exactement la même chose avec la même erreur à chaque copie. Nous devons considérer l'erreur multinomiale ($n_c = 1000$) en plus de l'erreur systématique, car l'erreur de QPT serait toujours nulle (ou de $\sim 10^{-8}$ car l'algorithme d'optimisation qui minimise $\mathcal{L}^{gauss}$ n'est pas parfait) si on ne le faisait pas.

Les erreurs systématiques sont dans un premier temps modélisées comme des erreurs qui ne changent pas le caractère non-intriqué de l'état, implicitement, on suppose donc que les états initiaux où tous les qubits sont initialisés à $|0\rangle$ sont bien non-intriqués et que les portes de Hadamard sont bien des portes unitaires mono-qubit (cela garantit qu'elles n'introduisent pas d'intrication). En pratique, pour modéliser cette erreur, nous utilisons la paramétrisation des états initiaux de (3.35) avec un r et un θ par qubit. En l'absence d'erreur systématique r vaut $\sqrt{2}$ si une porte de Hadamard a été appliquée au qubit et 1 sinon, θ vaut toujours 0. Nous modélisons l'erreur systématique comme une erreur gaussienne indépendante centrée sur tous les r et tous les θ d'écart-type σ_{sys} (exprimé en radians, que nous ferons varier) sur les θ et $\frac{\sigma_{sys}}{2\pi}$ pour les r. Si le r bruité sort de l'intervalle [0,1], alors nous lui soustrayons sa partie entière pour l'y ramener (dans ce cas, l'erreur n'est plus gaussienne). La figure 5.2 présente les boîtes à moustaches des erreurs de QPT en fonction de l'écart-type (std.) de l'erreur systématique. L'écart-type infini (inf) correspond à l'état initial totalement aléatoire.

La figure 5.2 montre les erreurs de l'estimée initiale ($\widehat{\mathbf{M}}_{LS}$) et de l'estimée de maximum de vraisemblance ($\widehat{\mathbf{M}}_{ML}$) pour un σ_{sys} qui varie linéairement de 0 à $0,3$ rad et pour $\sigma_{sys} = 100$ rad. Cette dernière valeur vise à modéliser des états initiaux totalement aléatoires et non-intriqués.

La précision des estimateurs est assez peu sensible à l'erreur systématique. C'est surtout vrai pour l'estimateur du maximum de vraisemblance. Pour l'estimateur initial, l'erreur augmente de façon non négligeable (surtout pour les quantiles d'ordre plus élevés) jusqu'à à peu près $\sigma_{sys} = 0.2$ rad, puis elle diminue légèrement. Cette augmentation est sans doute due à un conditionnement dégradé du problème pour des états initiaux proches des états cibles qui rend notre estimée initiale vulnérable à l'erreur multinomiale. Nous considérons quand même que la résistance aux erreurs systématique est satisfaisante, surtout pour l'estimateur du maximum de vraisemblance.

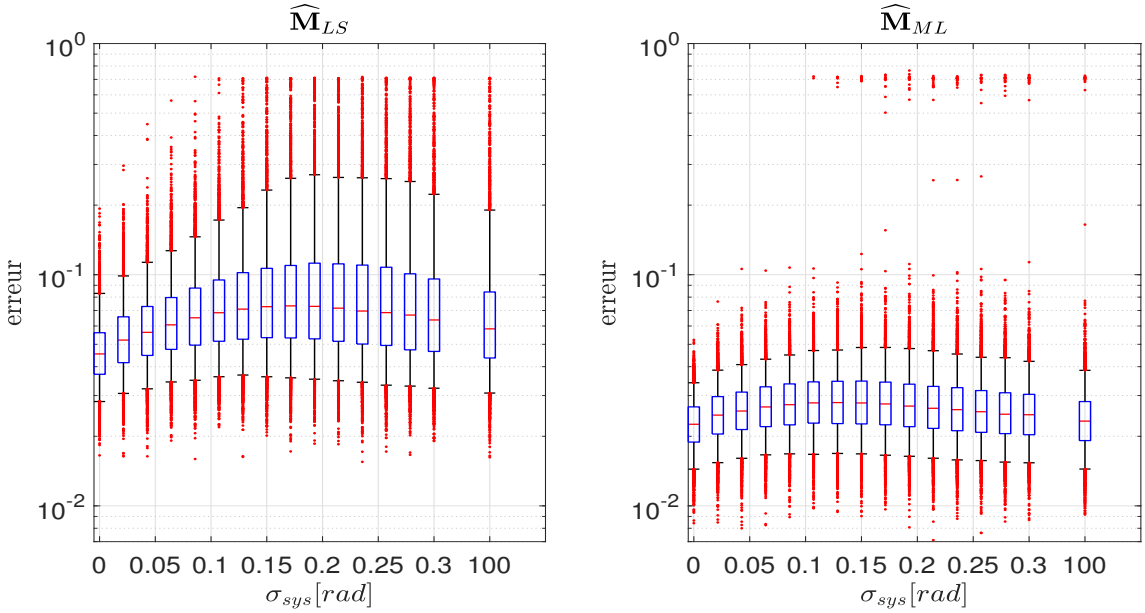


FIGURE 5.2 : Boîtes à moustaches des erreurs de QPT en présence d'erreurs systématiques et avec $n_c = 1000$ pour les deux algorithmes de QPT (l'initialisation qui donne $\widehat{\mathbf{M}}_{LS}$, le maximum de vraisemblance qui donne $\widehat{\mathbf{M}}_{ML}$).

5.1.3 Comparaison avec l'algorithme de Baldwin et al.

Dans la présente section, nous voulons comparer notre algorithme avec celui de [BKD14] dans le cas où il y a deux qubits et quatre états initiaux ($n_{qb} = 2, n_i = 4$). Ce n'est pas une tâche triviale car l'algorithme de [BKD14] a été conçu pour la configuration SQPT de la figure 5.3 (dans le cas $n_{qb} = 2, n_i = 4$). Notre algorithme a été défini pour la configuration semi-aveugle de la figure 5.4. Nous pouvons cependant l'adapter à la configuration SQPT de la figure 5.3 (voir la section 3.1.4), au prix de perdre notre résistance aux erreurs systématiques. Nous avons choisi de comparer les trois algorithmes suivants :

1. Notre algorithme d'initialisation fonctionnant sur la configuration de la figure 5.4 avec $n_c = 1000$.

2. L'algorithme de [BKD14] sur la configuration de la figure 5.3. Nous avons choisi de fixer $n_c = 2000$ afin de tenir compte du fait que la figure 5.3 exige que quatre états soient mesurés au lieu de huit.

3. Notre algorithme d'initialisation sur la configuration de la figure 5.3 avec les adaptations de la section 3.1.4 et avec $n_c = 2000$.

Nous ne testons pas l'algorithme du maximum de vraisemblance, car il fait l'hypothèse que les états initiaux sont non-intriqués. Nous pourrions le modifier pour l'adapter à des états initiaux intriqués (comme nous allons le faire dans la section 5.1.4), mais la comparaison avec l'algorithme de Baldwin et al. serait déloyale, car ce dernier n'est pas adapté pour compenser la présence d'erreur (multinomiale ou autre), alors que l'algorithme du maximum de vraisemblance prend beaucoup plus de temps pour compenser au mieux les erreurs. Pour ces trois options, les états initiaux sont les états définis dans l'équation (20) de [BKD14] ou dans (1.19) dans le présent manuscrit. Nous introduisons une erreur systématique intriquée (car les états sont intriqués), elle est modélisée comme une erreur gaussienne complexe centrée iid. sur toutes les composantes des états initiaux et d'écart-type variable. Les états bruités sont ensuite re-normalisés. Par souci

de simplicité, nous utilisons l'algorithme de QST de la section 2.3 pour tous les algorithmes QPT.

Comparons tout d'abord le graphique de gauche avec les deux autres graphiques de la figure 5.5. Lorsque l'erreur systématique augmente, l'erreur est plus faible dans le graphique de gauche. Cela n'est pas surprenant, puisque notre algorithme pour la configuration semi-aveugle (dont les performances sont représentées sur le graphique de gauche) a été conçu pour résister aux erreurs systématiques, il n'utilise donc pas les valeurs de l'état initial. En revanche, les deux autres algorithmes supposent que les valeurs des états initiaux sont exactement connues. Il est également logique que pour des erreurs systématiques plus faibles, les algorithmes non aveugles fonctionnant sur la configuration SQPT (ils sont représentés au milieu et à droite de la figure 5.5) soient plus performants. En effet, ces derniers utilisent les valeurs des états initiaux plutôt que de gaspiller la moitié des mesures pour obtenir une estimation bruitée des états $\mathbf{M}\boldsymbol{v}_j$ qui jouent le même rôle dans la configuration semi-aveugle que les états d'entrée $\boldsymbol{v}_j$ dans la configuration aveugle.

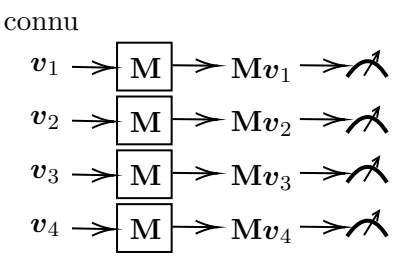


FIGURE 5.3 : Configuration de la SQPT avec 2 qubits.

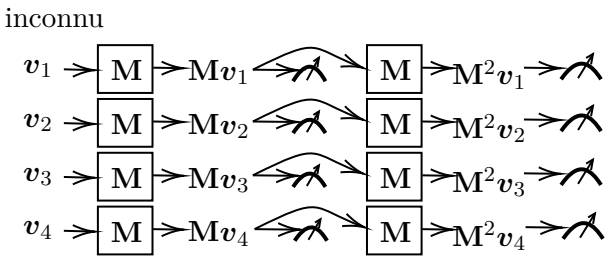


FIGURE 5.4 : Configuration semi-aveugle $n_i = 4, n_s = 2, n_{qb} = 2$.

Le graphique du milieu de la figure 5.5 (algorithme de [BKD14] appliqué à la configuration SQPT de la figure 5.3) présente des erreurs plus élevées que le graphique de droite pour toutes les valeurs (sauf pour la "std infinie", où elles sont identiques). Cela signifie que notre algorithme adapté à la configuration SQPT donne une meilleure estimation que celui de [BKD14]. Cela s'explique par le fait que ce dernier est très simple et élégant, mais qu'il n'a pas été conçu pour atténuer l'effet des erreurs que nous modélisons ici (erreur systématique et multinomiale). Il estime directement les coefficients de la matrice unitaire à partir des estimations QST des états. La matrice résultante n'a aucune raison d'être unitaire s'il y a des erreurs. Cela contraste avec notre algorithme de QPT qui trouve la matrice unitaire qui correspond le mieux à notre estimation de la QST (au sens des moindres carrés). L'algorithme de [BKD14] est plus simple et plus rapide, mais comme nous le verrons dans la section 5.1.8, notre algorithme d'initialisation de la QPT est vraiment rapide, au moins par rapport à l'algorithme de QST que nous utilisons (il est si rapide que son temps d'exécution est négligeable par rapport à la QST).

Dans l'ensemble, les performances de notre algorithme sur la configuration semi-aveugle sont très satisfaisantes. Il produit des erreurs plus faibles que l'algorithme de [BKD14] (resp. que notre algorithme avec les adaptations de la section 3.1.4) lorsque l'écart-type de l'erreur systématique gaussienne est d'environ $0,007$ (resp. $0,025$) sur chaque composante des états initiaux. Ces valeurs ($0,007$ et $0,025$) sont obtenues en interpolant linéairement les médianes des boîtes à moustaches dans les 3 graphiques et en calculant les valeurs des erreurs systématiques pour lesquelles les segments interpolés se croisent.

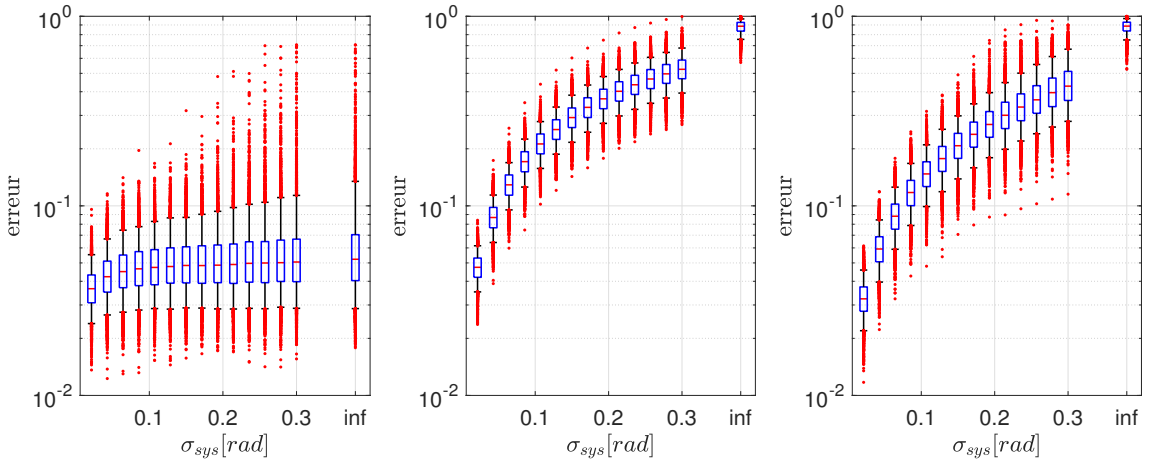


FIGURE 5.5 : Boîtes à moustaches de l'erreur de QPT en présence d'erreurs systématiques avec les états initiaux de [BKD14]. Le premier graphique (côté gauche) représente l'erreur avec notre algorithme fonctionnant dans la configuration en semi-aveugle de la figure 5.4. Le deuxième graphique (au milieu) représente l'erreur avec l'algorithme proposé dans [BKD14] sur la configuration standard considérée dans [BKD14] (représentée dans la figure 5.3 pour deux qubits). Le troisième graphique (partie droite) représente les performances de notre algorithme adapté pour fonctionner dans la configuration SQPT de la figure 5.3.

5.1.4 États d'entrée intriqués

Dans cette section, nous étudions les effets d'une erreur systématique qui intrique les états d'entrée. Nous avons été réticents à considérer l'intrication des états d'entrée plus haut parce que nous pensons que l'hypothèse de non intrication est raisonnable. En effet, si ce n'est pas le cas, c'est que (i) les états censés être préparés à $|0\rangle$ sont en réalité intriqués, ou alors (ii) que les portes mono-qubit de Hadamard créent une intrication entre les qubits. Nous pensons que l'opérateur peut considérer que les états sont non-intriqués car la bonne préparation d'états mono-qubit à $|0\rangle$ est la base du calcul quantique, et, si on est dans le cas (ii), alors, les portes de Hadamard réalisées ne sont pas unitaires (un état pur mono-qubit entre dans la porte, et un qubit intriqué avec le reste du circuit (il est donc un état mélange quand il est considéré tout seul) en sort), or l'hypothèse centrale de la présente thèse est que le processus à identifier est unitaire. Si on ne sait pas faire des portes mono-qubit unitaires, cette hypothèse n'est peut-être pas raisonnable. Ce dernier argument est à relativiser cependant, nous pensons qu'il est plus facile d'isoler le circuit à deux qubits du reste de l'environnement que d'isoler le premier qubit du deuxième qubit pendant la phase de la préparation des états alors qu'ils sont adjacents sur le circuit et qu'ils vont se faire intriquer par la porte à identifier à l'étape suivante.

Dans la présente section, nous levons donc cette hypothèse de non-intrication, nous modélisons l'erreur systématique avec des erreurs additives sur tous les coefficients des vecteurs d'états. Ces erreurs sont gaussiennes iid. centrées d'écart type σ_{ent} que nous faisons varier. Nous re-normalisons ensuite les états. Une telle erreur détruit la non-intrication des états d'entrée. L'algorithme qui calcule l'estimée initiale $\widehat{\mathbf{M}}_{LS}$ est adapté à ce type d'erreur, car il ne considère pas les états initiaux comme non intriqués. Ce n'est pas le cas pour l'estimateur du maximum de vraisemblance qui calcule $\widehat{\mathbf{M}}_{ML}$. Il cherche des états initiaux non intriqués et le processus unitaire qui maximisent la vraisemblance des mesures. Cet algorithme peut être adapté pour modéliser des états d'entrée intriqués cependant, il faut simplement changer la paramétrisation des états initiaux dans 3.3 avec celle des états de 2.4 (cette dernière est adaptée pour les états en sortie du processus à identifier, elle ne suppose donc pas que les états sont non intriqués). Nous

appelons $\widehat{\mathbf{M}}_{MLE}$ l'estimée qui en résulte. Si les états d'entrée sont non intriqués, elle devrait être moins précise que $\widehat{\mathbf{M}}_{ML}$ car un état intriqué a plus de paramètres qu'un état non intriqué, mais si les états d'entrée sont assez intriqués, $\widehat{\mathbf{M}}_{MLE}$ devrait être meilleure que $\widehat{\mathbf{M}}_{ML}$.

La figure 5.6 montre les boîtes à moustaches des erreurs des trois estimateurs ($\widehat{\mathbf{M}}_{LS}, \widehat{\mathbf{M}}_{MLE}$ et $\widehat{\mathbf{M}}_{ML}$) sur le circuit de la figure 3.4 avec $n_c = 1000$.

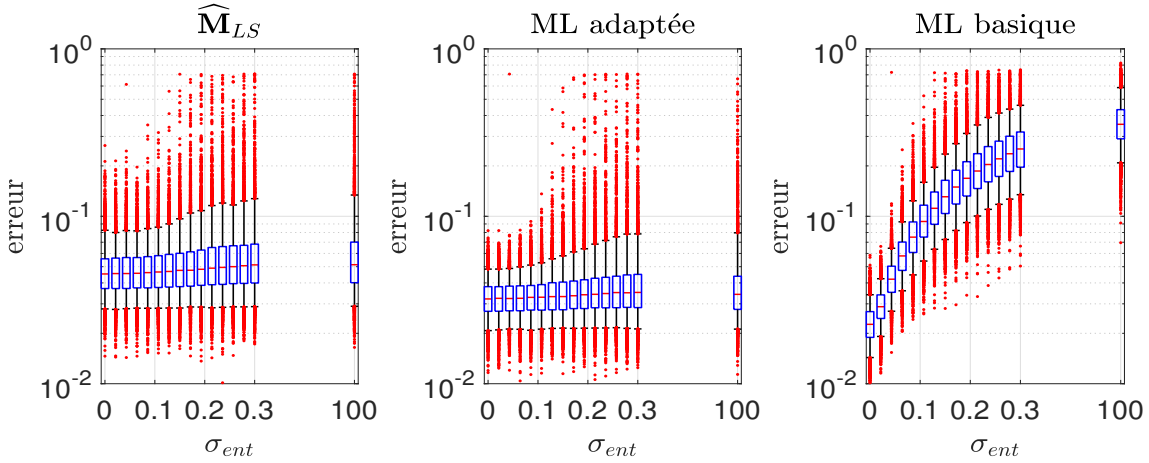


FIGURE 5.6 : Boîte à moustaches de l'erreur de QPT avec des états non-purs en entrée et avec $n_c = 1000$ avec des erreurs systématiques qui rendent les états initiaux intriqués. L'erreur est calculée pour l'estimateur initial $\widehat{\mathbf{M}}_{LS}$ à gauche, l'estimateur du maximum de vraisemblance adapté aux états intriqués $\widehat{\mathbf{M}}_{MLE}$ au milieu et l'estimateur du maximum de vraisemblance de base $\widehat{\mathbf{M}}_{ML}$ à droite.

Comme on pouvait s'y attendre, l'algorithme du maximum de vraisemblance de base (à droite) a des performances fortement dégradées quand σ_{ent} augmente. Cela contraste avec les performances des deux autres estimateurs ($\widehat{\mathbf{M}}_{LS}$ et $\widehat{\mathbf{M}}_{LSE}$) qui ne sont pas affectées de manière significative par l'augmentation de σ_{ent}. En pratique, les quantiles d'ordres plus élevés augmentent un peu pour ces deux estimateurs, mais cela est dû au fait que les états initiaux s'éloignent des états initiaux de (3.21) et pas au fait qu'ils deviennent intriqués (on observait le même phénomène avec les erreurs systématiques qui conservaient la non intrication sur la figure 5.2).

Si les états sont non intriqués ($\sigma_{ent} = 0$), l'estimateur du maximum de vraisemblance basique est plus précis que l'estimateur du maximum de vraisemblance adapté aux états intriqués, la médiane de l'erreur du premier est $0,0226$ dans notre exemple (50% plus faible que l'erreur d'initialisation) et la médiane du second est $0,0321$ (31% plus faible que l'erreur d'initialisation)

5.1.5 États mélange et processus non-unitaire

Les erreurs systématiques ne sont pas les seuls types d'erreurs pouvant survenir lors de la préparation de l'état d'entrée. Les autres types d'erreurs possibles sont les suivants :

- Préparation de l'état de base ($|0\rangle$) qui n'est pas la même sur chaque copie.
- Portes de Hadamard qui ne font pas la même chose sur chaque copie.
- Une décohérence se produit pendant l'initialisation.
- Une décohérence se produit après l'initialisation.

Toutes ces erreurs, sauf la dernière, peuvent être modélisées en considérant que les états d'entrée sont des états mélange. Le dernier type d'erreur implique que le processus à identifier n'est pas unitaire (la décohérence est non-unitaire par nature, si elle se produit après l'initialisation c'est qu'elle est générée par le processus que l'on veut identifier), elle sera testée à part.

Afin de modéliser les trois premiers types d'erreurs, nous simulons des états initiaux comme des états mélanges. Un état mélange est paramétré par une matrice $\boldsymbol{\rho} \in \mathbb{H}_d^+(\mathbb{C})$ de trace unitaire dont la décomposition spectrale est $\boldsymbol{\rho} = \sum_{k=1}^{d} p_k^{\boldsymbol{\rho}} \boldsymbol{v}_k^{\boldsymbol{\rho}} \boldsymbol{v}_k^{\boldsymbol{\rho}*}$, où les $\{p_k\}$ sont les valeurs propres (ordre décroissant) qui sont positives et dont la somme est égale à un, et $\{\boldsymbol{v}_k\}$ sont les vecteurs propres (voir section 1.1.8). Nous définissons $q_1 = 1 - p_1^{\boldsymbol{\rho}}$, cette quantité quantifie la proximité de l'état représenté par $\boldsymbol{\rho}$ par rapport à un état pur. Si $q_1 = 0$, alors l'état est pur et peut être représenté par $\boldsymbol{v}_1^{\boldsymbol{\rho}}$. La valeur maximale que q_1 peut prendre est $1 - \frac{1}{d}$. Si c'est le cas l'état représenté par $\boldsymbol{\rho}$ est l'entrée la moins utile que nous puissions considérer. Il est facile de vérifier que tout processus unitaire appliqué à cet état le laisse inchangé, et que toute mesure à d résultats possibles effectuée sur lui peut produire chaque résultat de manière équiprobable (probabilité $\frac{1}{d}$ pour chaque résultat).

Nous simulons des états non-purs avec des q_1 allant de 10^{-2} à 0,5. Les valeurs propres les plus petites sont toutes fixées à $\frac{1-q_1}{d-1}$. Nous testons les performances de notre algorithme de QPT sur ces états mixtes avec et sans l'erreur multinomiale. Les erreurs résultantes sont affichées sur la figure 5.7.

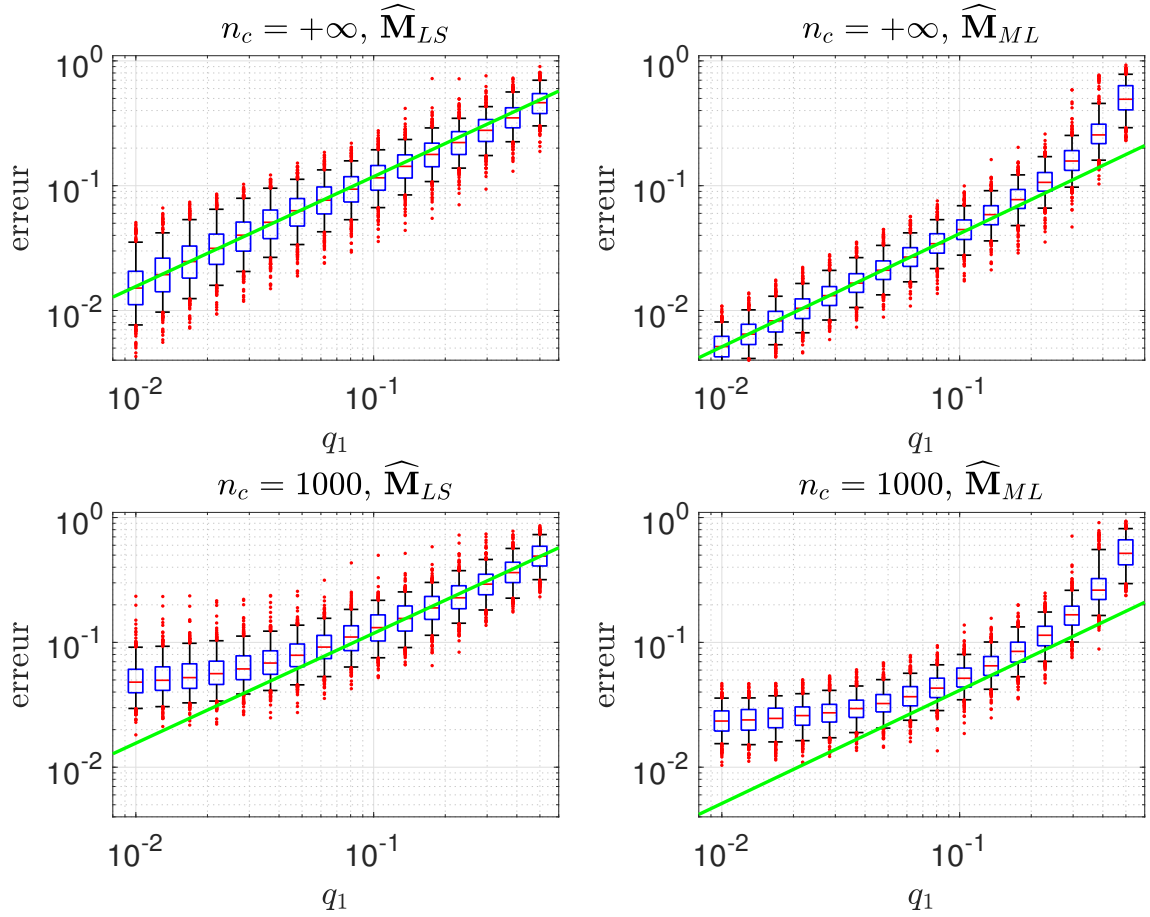


FIGURE 5.7 : Boîtes à moustaches de l'erreur de QPT avec des états d'entrée mixtes et avec $n_c = 1000$ et sans erreurs multinomiales ($n_c = +\infty$). Les lignes (vertes) allant du bas à gauche au haut à droite sur les graphiques sont calculées en ajustant le logarithme de la médiane de l'erreur (pour $q_1 < 0.1$) dans le cas $n_c = +\infty$, avec le logarithme de q_0 (il s'agit de la même ligne pour $n_c = +\infty$ et $n_c = 1000$, mais elles sont différentes pour l'estimée du maximum de vraisemblance et l'initialisation). La pente et le niveau de la ligne sont tous deux déterminés par une régression linéaire. Ceci diffère de la figure 5.1 où la pente a été fixée à $0,5$. Les deux pentes sont presque les mêmes (environ 0,9), les ordonnées à l'origine diffèrent, cette différence correspond à peu près à un facteur 3.

Pour les deux courbes en haut de la figure 5.7, les droites (vertes) qui traversent les boîtes à moustaches ont été calculées en effectuant une régression linéaire sur les médianes associées à des $q_1 < 0.1$ (la régression minimise le critère des moindres carrés sur les logarithmes). La pente trouvée par la régression est de 0,9, ce qui signifie que la médiane de l'erreur est à peu près proportionnelle à $q_0^{0,9}$. Cette pente change lorsque nous modifions les paramètres des matrices de densité simulées (elle dépend du rang et des valeurs relatives des plus petites valeurs propres), mais elle reste à peu près au même niveau, légèrement en dessous de 1.

Le gain en précision apporté par l'algorithme du maximum de vraisemblance est très important, surtout sans l'erreur multinomiale (l'erreur de QPT est divisée par 3 quand on passe de l'initialisation à ML sur les graphes pour lesquels $n_c = +\infty$). On constate également, que, pour la valeur d'erreur multinomiale que nous avons choisie ($n_c = 1000$), l'erreur de QPT générée par la non-pureté des états n'est presque pas visible pour $q_0 < 0.05$.

Testons maintenant nos estimateurs avec un processus à identifier qui n'est pas unitaire (il introduit de la décohérence après l'initialisation). Par souci de simplicité, on modélise le

processus ainsi :

$$\boldsymbol{\epsilon}(\boldsymbol{\rho}) = p_u \mathbf{M}\boldsymbol{\rho}\mathbf{M}^* + q_u tr(\boldsymbol{\rho})\mathbf{I}_d \tag{5.2}$$

où $\mathbf{M}$ est la matrice qui représente la partie unitaire du processus à identifier, p_u et q_u sont des réels entre 0 et 1 tels que $p_u = 1 - q_u$. Nous allons faire varier q_u. Il est trivial de vérifier que $\boldsymbol{\epsilon}(\boldsymbol{\rho})$ est linéaire et préserve la positivité et la trace. Pour chaque valeur de q_u, nous testons nos algorithmes avec les circuits de la figure 3.4 et de la figure 3.5[1] pour 500 processus non-unitaires différents (correspondant à des $\mathbf{M}$ générés aléatoirement) puis nous calculons l'erreur de (5.1) entre $\mathbf{M}$ (on considère que $\mathbf{M}$ représente le processus unitaire que l'on cherche et que le reste de $\boldsymbol{\epsilon}$ est du bruit).

Nous simulons ce processus avec le protocole de la figure 3.4 avec $n_{qb} = 2, n_i = 4$ et $n_s = 5$ (comme pour presque tous les tests précédents) et celui de la figure 3.5 avec $n_i = 1$ et $n_s = 5$. Cette dernière configuration est testée ici (et pas dans les sections précédentes) car le processus non-unitaire fait que plus on observe l'état longtemps plus on s'écarte du modèle d'état pur. Il est donc pertinent de tester les performances avec $n_s = 2$ et $n_s = 5$ pour avoir un n_s différent de 2 qui correspond à un temps d'observation plus long, les performances devraient donc être plus mauvaises. L'erreur de la QPT est affichée avec les deux algorithmes de QPT pour n_c fini et $n_c = +\infty$ et pour $n_i = 4$ (sous entendu avec la configuration de la figure 3.4, $n_s = 2$) et $n_i = 1$ (sous entendu avec la configuration de la figure 3.5, $n_s = 4$). Pour comparer de façon équitable les configurations $n_i = 1$ et $n_i = 4$ il faut que le nombre de copies total préparé $n_{tot} = n_i n_s n_t n_c$ soit le même dans les deux cas. On prend donc $n_c = 1000$ pour la configuration de la figure 3.4 avec $n_i = 4$, et on prend donc $n_c = 1600$ pour la configuration de la figure 3.5 avec $n_i = 4$. Les résultats sont présentés en figure 5.8.

Comme on s'y attendait, la configuration de la figure 3.4 ($n_i = 4$) est plus adaptée à la non-unitarité du processus (décohérence) que celle de la figure 3.5 ($n_i = 1$). Dans les deux cas, pour $n_c = +\infty$ les médianes des erreurs sont à peu près proportionelles à $q_u^{1,5}$ (coefficient directeur proche de 1,5 sur le graphe log-log) de q_u pour des valeurs raisonnables de q_u ($q_u < 0{,}25$) mais le coefficient de proportionnalité (ordonnée à l'origine de la droite en log-log) est plus grand (d'un facteur d'à peu près 2,7) pour $n_i = 1$ que pour $n_i = 4$. La différence entre les deux configurations est moins marquée quand n_c est fini, car l'erreur "multinomiale" prend le dessus pour les faibles q_u, mais elle reste présente.

[1] Il covient de noter que ces figures ont étées crées pour un processus unitaire. On peut cependant imaginer que le processus représenté par $\mathbf{M}$ est non unitaire. Dans ce cas (que nous considérons ici), les états d'entrée ($\boldsymbol{v}_k$) sont bien purs, mais les états suivants (notés $\mathbf{M}\boldsymbol{v}_k$) sont des états mélanges.

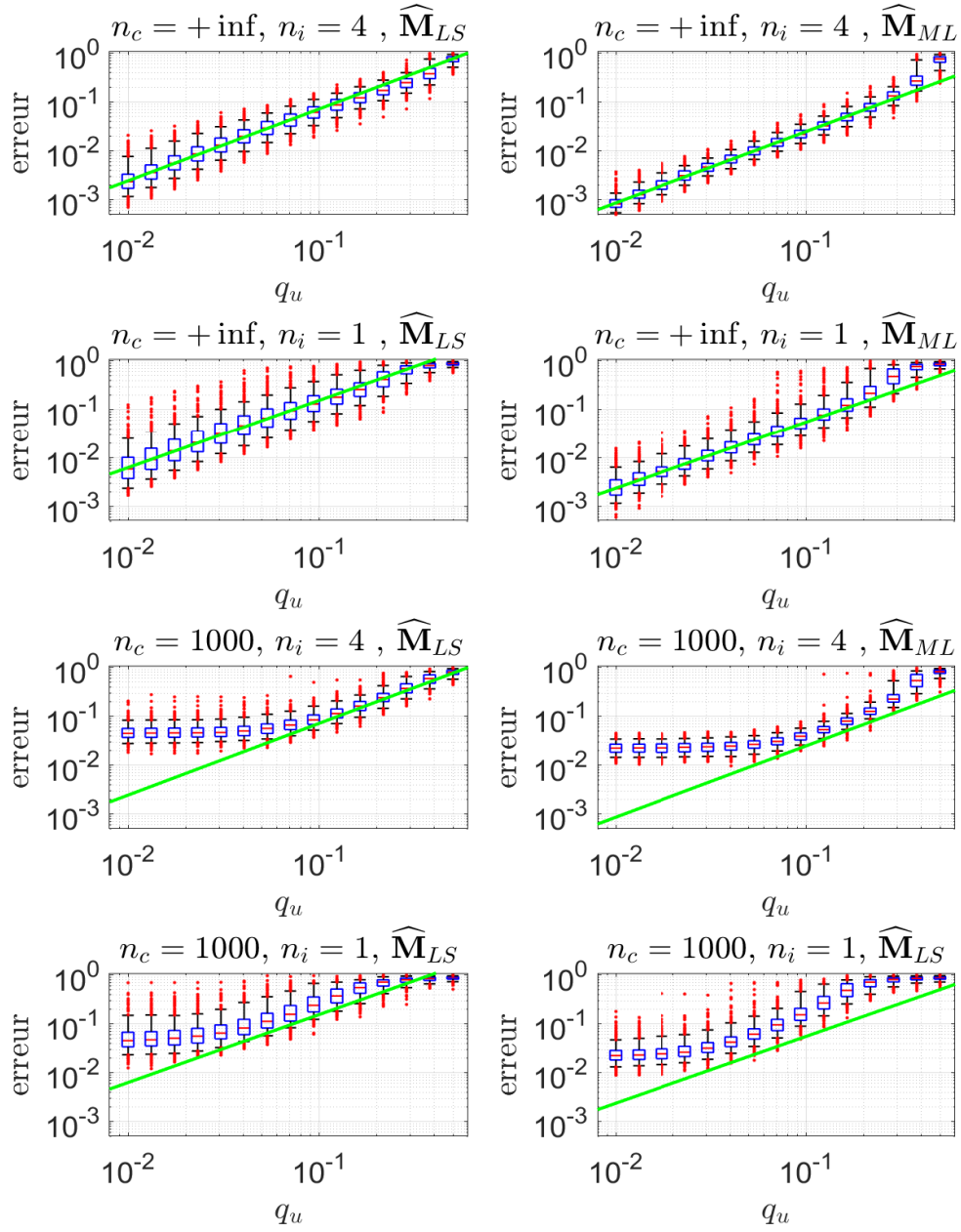


FIGURE 5.8 : Erreur des deux algorithmes de QPT avec le protocole de la figure 3.4 ($n_i = 4$) et celui de la figure 3.5 ($n_i = 1$), avec n_c fini et infini (n_{tot} reste constant). Les lignes vertes sont calculées sur les 4 graphes d'en haut (régression linéaire des médianes des erreurs associées à un q_u inférieur à $0,1$ sur les graphes en log-log) avec $n_c = +\infty$, et recopiées sur les 4 graphes correspondants de la partie inférieure avec n_c fini.

5.1.6 Erreurs de mesures et QMT

Dans cette section, nous testons la résilience de nos algorithmes de QPT aux imperfections du modèle des mesures, nous évaluons aussi notre algorithme de QMT du chapitre 4. Comme expliqué dans la section 4.4, notre algorithme de QMT peut estimer les paramètres des mesures avec les mesures en sortie d'un circuit de QPT si l'on utilise un circuit alternatif du type de la figure 3.2 qui mesure les états initiaux non intriqués (contrairement au circuit de base de la figure 3.1). Nous avons donné l'exemple du circuit de la figure 4.3 qui crée des états initiaux qui rendent la QMT la plus précise possible. Nous allons tester les performances de nos algorithmes pour le circuit de la figure 4.3 et les comparer aux performances pour le circuit de la figure 5.9 (le même que le circuit de la figure 4.3 mais avec des portes de préparation plus classiques (Hadamard et identité) qui permet aussi de réaliser la QMT avec les mesures de la QPT, mais avec des états initiaux non-mesurés.

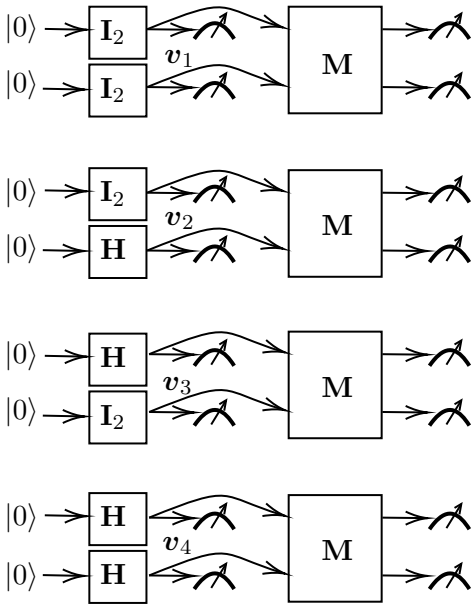


FIGURE 5.9 : Circuit alternatif pour réaliser la QMT avec les mesures de la QPT. Il s'agit d'une modification du circuit de la figure 3.4 avec $n_{qb} = 2$ et avec les états initiaux mesurés avant et pas après l'application de la porte à identifier (représentée par $\mathbf{M}$).

Nous simulons les configuration des figures 4.3 et 5.9 où chaque état mesuré est mesuré avec les $n_t = 4$ types de mesures de la section 2.2 [2]. Nous modélisons des mesures non intriquées, mais non conformes au modèle. Par exemple, pour la mesure ZZ, le modèle des mesures prévoit que la matrice des vecteurs propres est $\mathbf{I}_2 \otimes \mathbf{I}_2 = \mathbf{I}_4$ mais, à la place, on simule une vraie matrice de vecteur propre associée à la vraie mesure ZZ (que l'on ne connait pas pour la QPT) qui vaut $(\mathbf{I}_2\mathbf{P}_1^Z) \otimes (\mathbf{I}_2\mathbf{P}_2^Z)$. Les matrices $\{\mathbf{P}_j^T\}_{j\in 1,2,T\in\{X,Y,Z\}}$ sont des matrices unitaires aléatoires tirées de façon indépendante sur chaque qubit pour les trois types de mesures. Elles s'écrivent $\begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r)e^{i\phi_r} \\ \sin(\theta_r) & \cos(\theta_r)e^{i\phi_r} \end{pmatrix}$ où θ_r et ϕ_r sont des variables aléatoires gaussiennes centrées i.i.d. dont on fera varier l'écart type σ_r. Nous choisissons un nombre de mesures par état et par type de mesure grand ($n_c = 2500$) pour nous assurer que la correction apportée par la QMT soit plus

[2] C'est la seule simulation du présent chapitre qui n'utilise pas les $2n_{qb} + 1$ types de mesures de la section 2.3. Nous faisons ce choix car, comme expliqué dans la section 3.3.3, on doit utiliser des types de mesures qui font que chaque qubit est mesuré selon X, Y et Z

apparente sur les graphes des erreurs de QPT.

La figure 5.10 représente les boîtes à moustaches de l'erreur de QPT avec QMT (à droite) et sans QMT (à gauche), et avec les configurations de la figure 5.9 (en haut) et de la figure 4.3 (en bas) pour l'estimateur du maximum de vraisemblance avec la version gaussienne de la vraisemblance. L'erreur sur le processus est calculée dans la base de référence définie à partir des mesures (voir section 4.2).

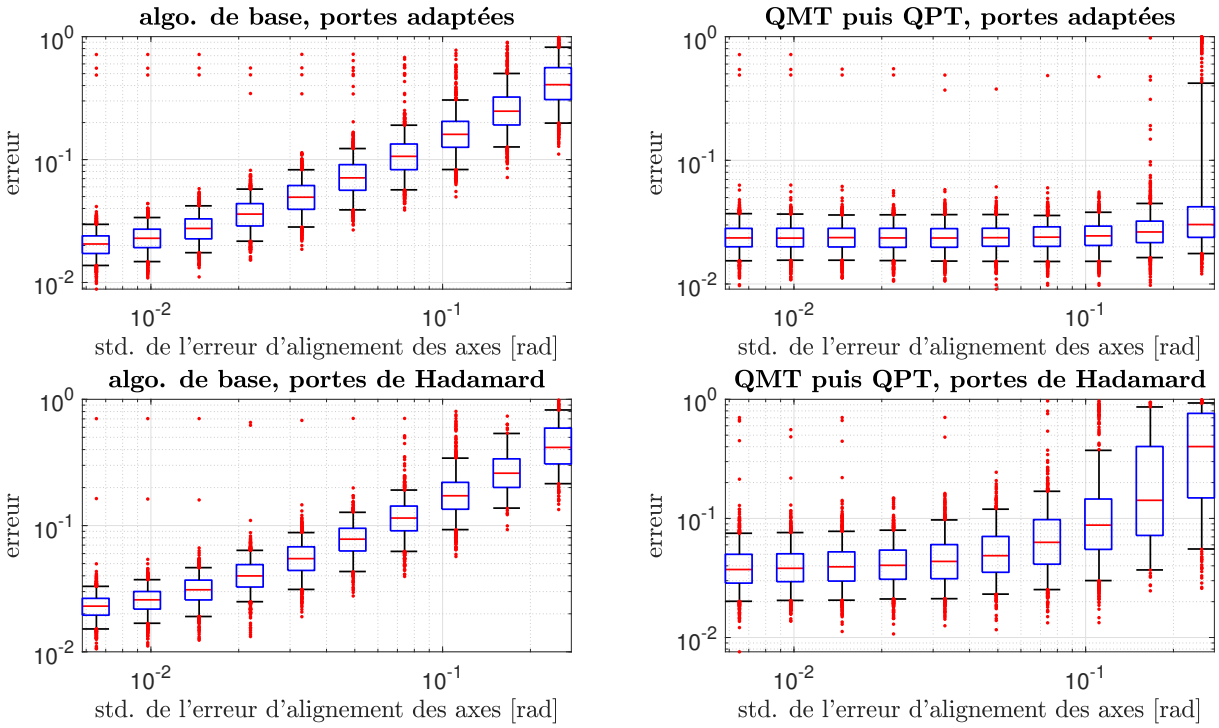


FIGURE 5.10 : Résultats de la QPT avec et sans QMT en présence d'erreurs sur le modèle des mesures, et avec deux types de portes de préparation d'états d'entrée différents.

Sans appliquer l'algorithme de QMT (deux graphes de gauche), les précisions de la QPT sont comparables, avec un petit avantage pour la configuration de la figure 4.3 (adaptée à la QMT). Les erreurs sur le modèle des mesures sont subies, et, pour $\sigma_r \geq 2 \times 10^{-2}$, l'erreur de QPT semble croître de façon à peu près linéairement en fonction de l'écart type (en pratique cette croissance doit s'arrêter quand σ_r dépasse une certaine valeur que nous n'observons pas car l'erreur de QPT est bornée). En appliquant l'algorithme de QMT (deux graphes de droite), l'impact des erreurs sur le modèle des mesures est mieux maitrisé, et c'est particulièrement vrai pour la configuration de la figure 4.3, ce qui est logique puisqu'elle est adapté à la QMT. Avec cette configuration , on est insensible aux erreurs sur le modèle de mesure avec des écarts types sur les θ_r et ϕ_r inférieurs à 0,2 (ce qui est très grand). On a par contre une perte de performance pour les $\sigma_r < 0{,}01$ (cette borne aurait été inférieure si on avait considéré un n_c encore plus grand). C'est logique, la QMT est inutile si l'erreur sur le modèle de mesure est négligeable, et elle est même contre-productive, car elle remet en cause le modèle de mesure pour ré-estimer ses paramètres. Sans le circuit adapté, la QMT perd beaucoup de son intérêt : pour le circuit de la figure 4.3 (portes de Hadamard), les résultats sont bien moins bons que pour le circuit de la figure 5.9, et la plage de valeurs de σ_r sur laquelle la QPT améliore les résultats est considérablement réduite.

On peut isoler les performances de la QMT en affichant seulement les valeurs absolues des

erreurs d'estimation des angles θ_x, θ_y et ϕ_y qui paramétrisent des mesures X et Y qui sont estimés (voir section 4.2). Les boîtes à moustaches de ces angles pour les différentes valeurs de σ_r et pour les deux circuits sont fournies en figure 5.11 :

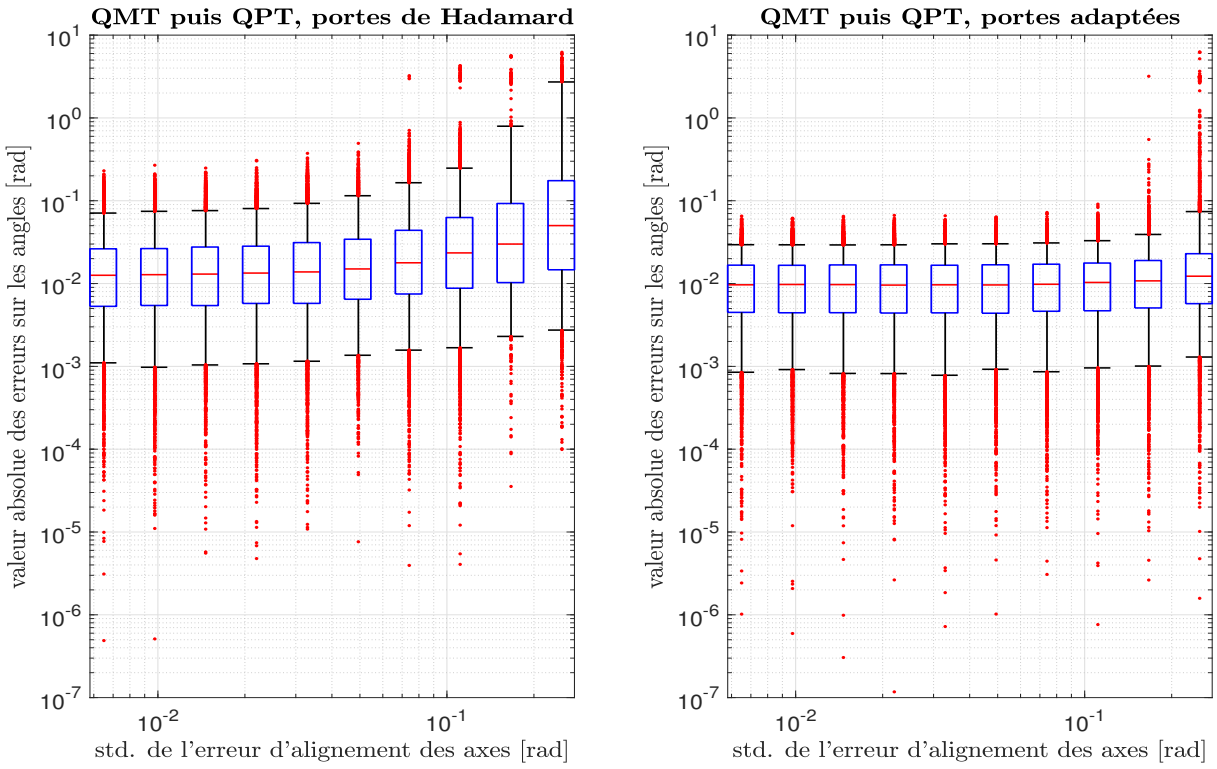


FIGURE 5.11 : Erreurs sur les angles estimés par la QMT.

Comme on pouvait s'y attendre, les erreurs sont plus faibles avec le circuit adapté à la QMT. C'est particulièrement flagrant sur les grands quantiles (3-ème quartile, 95-ème centile et outliers) et quand σ_r augmente. La médiane des erreurs des angles estimés vaut à peu près $1{,}0 \times 10^{-2}$, et, en pratique, elle est proportionnelle à $\frac{1}{\sqrt{n_c}}$.

5.1.7 Borne de Cramér-Rao

Dans cette section, nous vérifions que la borne de Cramér-Rao définie dans la section 3.3.5 permet bien de donner une bonne estimée de la matrice de covariance des parties réelles et imaginaires des colonnes de la matrice $\widehat{\mathbf{M}}_{ML}$ rephasée. Dans (3.44) nous avons une estimée de cette matrice de covariance calculée à partir de la borne de Cramér Rao. Cette dernière dépend des vrais paramètres (paramètres de la vraie matrice $\mathbf{M}$ et des vrais états initiaux), mais elle peut être estimée en remplaçant ces paramètres par les paramètres estimés en sortie de l'algorithme du maximum de vraisemblance.

Nous testons la validité de l'estimée de (3.44) avec une simulation pour deux qubits, avec une matrice de rotation aléatoire et les états initiaux aléatoires proches des états de (3.21). À partir de cette configuration fixée, nous avons des probabilités théoriques pour chacun des résultats possibles de chaque type de mesure fait pour chaque état. Nous simulons ensuite 4000 réalisations du modèle multinomial suivant les probabilités théoriques avec $n_c = 250$ réalisations (tout cela pour une seule matrice $\mathbf{M}$ aléatoire).

Pour chacune de ces 4000 réalisations, nous calculons l'estimée $\widehat{\mathbf{M}}_{ML}$ associée que nous trans-

formons en un vecteur de réels $\widehat{\boldsymbol{m}}_{ML}$ de taille $2d^2 = 32$ qui contient les d^2 parties réelles puis les d^2 parties imaginaires des colonnes de $\widehat{\mathbf{M}}_{ML}$ concaténées les unes sous les autres. Exceptionnellement, ici, $\widehat{\mathbf{M}}_{ML}$ est calculée en minimisant $\mathcal{L}^{exact}$ et non $\mathcal{L}^{gauss}$ car on veut (dans un premier temps) tester la borne de Cramér-Rao en la comparant à la covariance du maximum de la bonne vraisemblance, pas à la covariance d'un maximum d'une vraisemblance gaussienne régularisée que l'on a crée pour être plus facilement optimisable et modéliser d'autres types d'erreurs. On calcule ensuite la matrice de covariance empirique des $\widehat{\boldsymbol{m}}_{ML}$ que l'on compare avec la version théorique de (3.44) :

$$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{m}}}(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{M}}_{ML}) = \mathbf{J}_{reph}(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{M}}_{ML}) \mathbf{J}_{\mathbf{m}}(\widehat{\boldsymbol{\theta}}) \boldsymbol{\Sigma}_{\mathbf{h}}^{cr}(\widehat{\boldsymbol{\theta}}) \mathbf{J}_{\mathbf{m}}(\widehat{\boldsymbol{\theta}})^{*} \mathbf{J}_{reph}(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{M}}_{ML})^{*} \tag{5.3}$$

où les matrices $\mathbf{J}_{reph}(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{M}}_{ML})$, $\mathbf{J}_{\mathbf{m}}(\widehat{\boldsymbol{\theta}})$ et $\boldsymbol{\Sigma}_{\mathbf{h}}^{cr}(\widehat{\boldsymbol{\theta}})$ sont définies dans la section 3.3.5. Elles sont calculées en des points connus avec le dernier maximum de vraisemblance calculé : $\widehat{\mathbf{M}}_{ML}$ est l'estimée de $\mathbf{M}$ avec le maximum de vraisemblance et $\widehat{\boldsymbol{\theta}}$ sont les paramètres associés. Avec cette définition, $\boldsymbol{\Sigma}_{\widehat{\boldsymbol{m}}}(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{M}}_{ML})$ peut être calculée avec la dernière des 4000 réalisations des mesures multinomiales simulées (le choix de prendre la dernière réalisation est arbitraire). On appelle la matrice $\boldsymbol{\Sigma}_{\mathbf{h}}^{cr}(\widehat{\boldsymbol{\theta}})$ la borne de Cramér-Rao de l'estimée des paramètres de la matrice unitaire (c'est un abus de langage, la borne de Cramér-Rao concerne les paramètres $\widehat{\boldsymbol{\theta}}$, et pas les coefficients de $\widehat{\mathbf{M}}_{ML}$). Elle peut être définie avec le modèle gaussien ou le modèle multinomial (voir (3.41)). Dans un premier temps, nous choisissons de prendre le modèle multinomial, car il s'agit du modèle avec lequel nous avons généré les mesures et du modèle de la vraisemblance maximisée ($\mathcal{L}^{exact}$). La figure 5.12 représente les valeurs des coefficients de la matrice $\boldsymbol{\Sigma}_{\widehat{\boldsymbol{m}}}(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{M}}_{ML})$ (à gauche), de la matrice de covariance empirique (au milieu) et de l'erreur (à droite) :

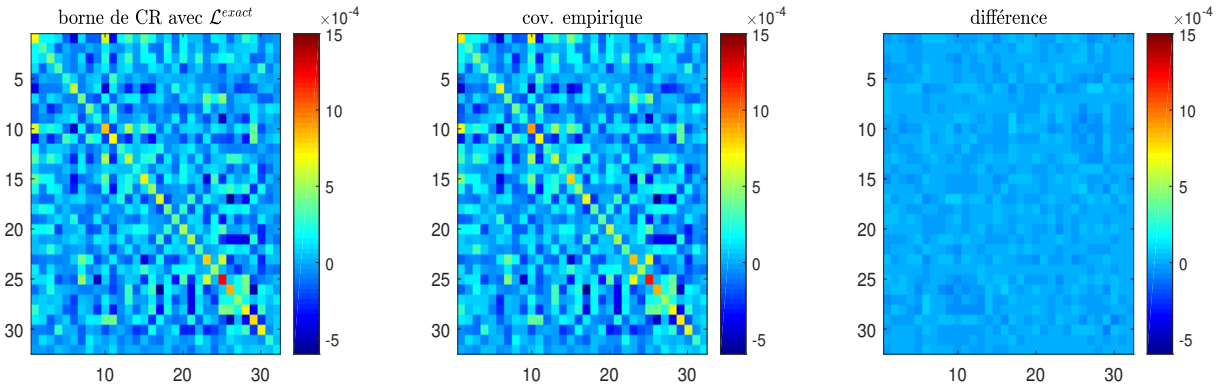


FIGURE 5.12 : Covariance de la concaténation des parties réelles et imaginaires de l'estimée des coefficients de la matrice unitaire normalisée. Indices des colonnes en abscisse, indices des lignes en ordonnée, la valeur des coefficients est donnée par la couleur. Version théorique (calculée à partir de la borne de Cramér-Rao) calculable sans références en sortie de l'algorithme du maximum de vraisemblance à gauche, version empirique de la covariance au milieu, écart entre les deux à droite.

La version théorique de la matrice de covariance calculée à partir de la borne de Cramér-Rao et la version empirique sont presque identiques. On peut calculer l'erreur relative entre la borne de Cramér-Rao et la covariance empirique (norme de la différence divisée par la norme de la covariance empirique) : elle vaut $0,1332$ et les plus grosses erreurs relatives sont hors de la diagonale (qui contient les variances), l'erreur relative sur la diagonale vaut $0,0462$.

On peut faire le même calcul avec la version gaussienne de la vraisemblance. Les résultats sont en figure 5.13. Il y a deux changements entre la simulation de la figure 5.12 et celle de la figure

5.13 : (i) la borne de Cramér-Rao est calculée à partir de la vraisemblance gaussienne (deuxième expression de (3.41) pour l'information de Fisher dont la borne de Cramér-Rao est l'inverse), et non la vraisemblance multinomiale, et (ii) la covariance empirique est calculée avec les $\widehat{\mathbf{M}}_{ML}$ qui minimisent la version gaussienne de la vraisemblance (et non la version multinomiale).

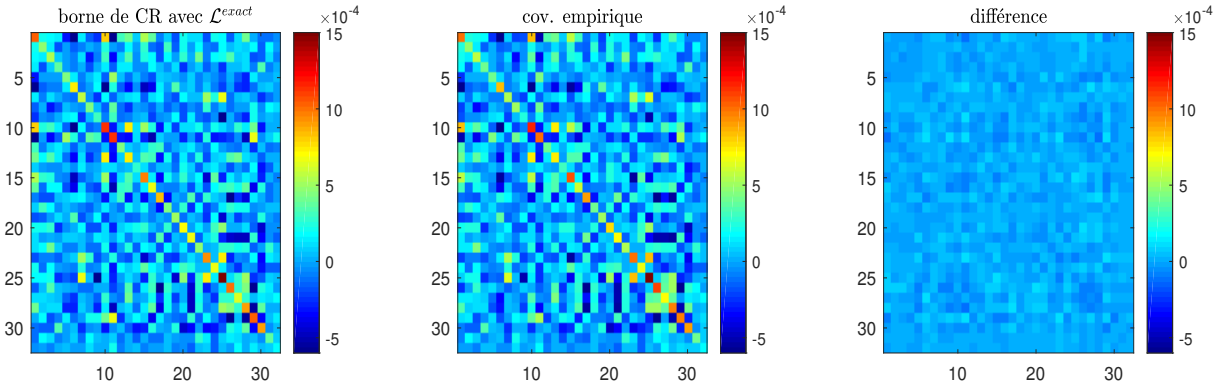


FIGURE 5.13 : Même légende que pour la figure 5.12, mais avec (i) une borne de Cramér-Rao calculée avec la version gaussienne régularisée de la vraisemblance (alors que les mesures sont générées avec le modèle multinomial), et (ii) la covariance empirique est estimée à partir des $\widehat{\mathbf{M}}_{ML}$ qui maximisent la version gaussienne de la vraisemblance.

L'erreur relative entre la borne de Cramér-Rao théorique et la covariance empirique est 0,1576 sur la figure 5.13 (0,0674 sur la diagonale). Cette erreur est légèrement supérieure à celle de la figure 5.12, car le modèle gaussien de la vraisemblance utilisé pour la borne de Cramér-Rao ne correspond pas à la distribution des mesures (multinomiales) avec laquelle on génère les mesures.

Il convient de noter que la variance de l'erreur théorique avec le modèle gaussien sur la figure 5.13 (somme des éléments de la diagonale de la matrice de covariance théorique (à gauche)) est supérieure à la variance de l'erreur théorique sur la figure 5.12 (modèle multinomial) : la variance théorique vaut 0,0219 pour le modèle gaussien 0,0167 pour le modèle multinomial. C'est assez intuitif, le modèle gaussien des mesures est plus entropique (plus les mesures sont entropiques plus on s'attend à ce que l'erreur soit grande) que le modèle multinomial car une densité continue a une entropie maximale (à variance constante) si elle est gaussienne, et le modèle gaussien augmente les variances des petites erreurs avec la régularisation.

Nous avons donc montré que, même avec un nombre assez faible de mesures ($n_c = 250$), on peut facilement estimer la covariance de l'erreur des coefficients de $\widehat{\mathbf{M}}_{ML}$ en approximant la borne de Cramér-Rao avec les paramètres estimés en sortie de l'algorithme de maximum de vraisemblance. Il ne faut pas oublier que la borne de Cramér-Rao dépend du modèle de la vraisemblance, comme on l'a vu, la variance théorique de l'erreur sur la matrice de rotation calculée avec le modèle gaussien est nettement supérieure à la variance calculée avec le modèle gaussien. Cependant, même avec un modèle de vraisemblance qui ne correspond pas tout à fait au modèle des mesures (inconnu en pratique), la borne de Cramér-Rao semble être une bonne estimation de la matrice de covariance empirique tant que la vraisemblance que l'on maximise est la même que la vraisemblance que l'on dérive pour avoir l'information de Fisher. En effet, l'écart entre les covariances théoriques est empirique sur la figure 5.13 n'est pas bien plus grand que celui sur la figure figure 5.12.

Toute l'information sur l'erreur n'est pas forcément contenue dans la matrice de covariance cependant. Les biais des estimées de chaque coefficient de la matrice unitaire estimée sont

aussi importants. Ces biais sont définis comme la version vectorisée de la différence entre la moyenne empirique des écarts entre les $\widehat{\mathbf{M}}_{ML}$ estimés par maximum de vraisemblance et la vraie matrice unitaire $\mathbf{M}$ rephasée par rapport à la dernière matrice $\mathbf{M}_{ML}$ calculée, c'est-à-dire : $\boldsymbol{f}_{vect}\left(\mathbf{M}\frac{tr(\mathbf{M}^*\mathbf{M}_{ML})}{|tr(\mathbf{M}^*\mathbf{M}_{ML})|}\right)$. La somme (sur les d^2 coefficients de $\mathbf{M}$) des variances empiriques des coefficients est plus de 300 fois supérieure à la somme des carrés des biais empiriques (sur la simulation qui a généré la figure 5.12).

On peut donc considérer que l'estimateur du maximum de vraisemblance n'est pas biaisé et que sa matrice de covariance est facilement estimable à partir des résultats de la maximisation de la vraisemblance $\widehat{\mathbf{M}}_{ML}$ et $\widehat{\boldsymbol{\theta}}$.

5.1.8 Plus de deux qubits

Maintenant que nous avons étudié l'impact des différents types d'erreurs, nous voulons voir si notre algorithme QPT fonctionne bien avec plus de deux qubits et comparer les configurations des figures 3.4 et 3.5.

Pour 1, 2, 3, 4, 5 et 6 qubits, nous simulons des 300 matrices unitaires aléatoires en appliquant le processus de Gram Schmidt à des matrices gaussiennes aléatoires. Nous essayons d'identifier les processus associés à chaque matrice avec les deux configurations suivantes :

1. Avec les $n_i = d$ états initiaux de (3.22) et $n_s = 2$ retards temporels de la figure 3.4. Nous simulons $n_c = 2\ 500$ mesures par type de mesure et par état mesuré. Nous considérons que les états sont préparés avec la configuration de la figure 3.5. Les portes de Hadamard utilisées pour la préparation de l'état initial sont considérées comme imparfaites et sont représentées par $\begin{pmatrix}\cos(\theta_r) & -\sin(\theta_r)e^{i\phi_r}\\ \sin(\theta_r) & \cos(\theta_r)e^{i\phi_r}\end{pmatrix}\mathbf{H}_d$ au lieu de $\mathbf{H}_d$, où θ_r et ϕ_r sont deux angles aléatoires i.i.d. gaussiens centrés avec un écart type de $0,05$ radians (deux valeurs différentes sont tirées pour chaque porte de la figure 3.5 et pour chacune des 300 portes à identifier).

2. Avec l'unique ($n_i = 1$) état initial aléatoire et les $n_s = d + 1$ retards temporels de la configuration de la figure 3.5 (mais avec un état initial aléatoire). Nous simulons $n_c = [2500\frac{2d}{d+1}]$ ($[x]$ est l'entier le plus proche du nombre réel x), de sorte que le nombre total de mesures $n_c n_i n_s n_t$ est le même (ou presque le même si $2500\frac{2d}{d+1}$ n'est pas un entier) sur les deux configurations pour un nombre donné de qubits.

Comme nous l'avons expliqué dans la section 3.2.3, la première configuration est plus robuste et la seconde peut être plus facile à préparer et permet de travailler avec des n_c plus élevés (car $\frac{2d}{d+1} > 1$).

Les valeurs n_c choisies sont vraiment élevées pour les petites valeurs de n_{qb}. Si $n_{qb} = 2$, par exemple, il n'y a que $d = 4$ résultats possibles dont nous voulons estimer les probabilités. Dans ce cas, $n_c = 2500$ ou $n_c = \left[2500\frac{2\times4}{4+1}\right] = 4000$ est plus que suffisant pour obtenir de très bonnes estimations. Mais si $n_{qb} = 6$, il y a $d = 64$ issues, et les probabilités associées sont beaucoup plus petites (puisqu'elles somment à un), alors le nombre de mesures ne semble pas si excessif.

La figure 5.14 montre les boîtes à moustaches pour les deux configurations quand le nombre de qubits varie, pour l'algorithme d'initialisation et pour l'algorithme de maximum de vraisemblance. Pour les deux algorithmes et pour tous les qubits, il a 2 boîtes à moustaches côte-à-côte, elles correspondent aux deux configurations de la liste ci dessus (configuration 1 à gauche et configuration 2 à droite).

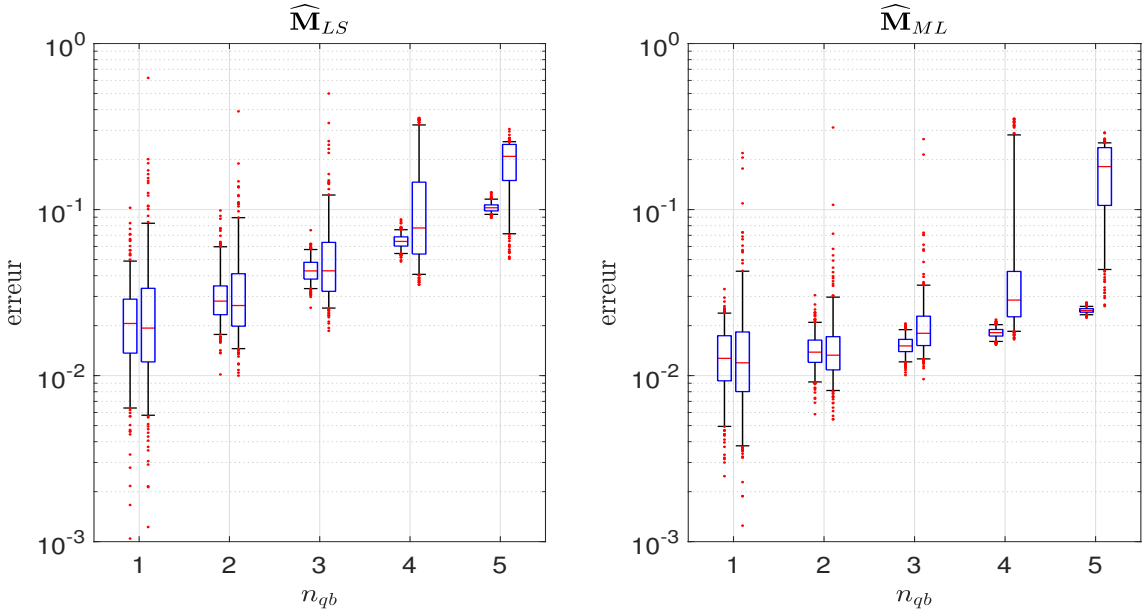


FIGURE 5.14 : Boîtes à moustaches des erreurs de QPT pour l'algorithme initial (qui donne $\widehat{\mathbf{M}}_{LS}$) et l'algorithme de maximum de vraisemblance (qui donne $\widehat{\mathbf{M}}_{ML}$) sur une porte quantique qui agit sur 1 à 5 qubits. Il y a deux boîtes à moustaches pour chaque nombre de qubits. Celles de gauche représentent les erreurs avec la première configuration ($n_i = d$, $n_s = 2$) et celles de droite représentent les erreurs avec la deuxième configuration ($n_i = 1$, $n_s = d + 1$). Les deux configurations ont le même nombre total de mesures.

Sans surprise, l'erreur augmente avec le nombre de qubits. C'est le cas bien que le nombre total de mesures $n_c n_i n_s n_t$ augmente également avec le nombre de qubits (n_c est constant mais $n_i = d$ ou 1 augmente pour la première configuration, $n_s = 2$ ou $d+1$ augmente pour la deuxième configuration et $n_t = 2n_{qb}+1$ augmente pour les deux configurations). Ceci n'est pas surprenant car le nombre de paramètres $d^2 = 2^{2n_{qb}}$ augmente exponentiellement avec le nombre de qubits.

Comparons dans un premier temps les erreurs dans la configuration de la figure 3.4 (boites à moustaches de gauche sur la figure 5.14) avec l'erreur dans la configuration de la figure 3.5 (boites à moustaches de droite). Pour la première configuration, la plage des valeurs prises par l'erreur diminue (en échelle logarithmique) lorsque le nombre de qubits augmente. Nous pensons que cela est dû au fait que, lorsque le nombre de composantes (de $\mathbf{M}$) à estimer augmente, l'erreur sur la matrice devient plus prévisible car les erreurs sur les composantes sont "moyennées" par la norme en (5.1). Nous n'observons pas ce phénomène avec la deuxième configuration car, sur les 300 simulations sur lesquelles les boîtes à moustaches de l'erreur sont calculées, le conditionnement de $\mathbf{X}$ et l'orthogonalité de ses colonnes varient beaucoup plus pour la deuxième configuration. Or ces deux quantités sont importantes pour la QPT (voir la fin de la section 3.2). Elles ne varieraient pas du tout pour la première configuration si nous n'avions pas introduit la petite perturbation aléatoire sur les portes de Hadamard.

Au delà de la plage des valeurs prises par l'erreur, les valeurs dans les deux configurations sont tout autant intéressantes. Pour un à quatre qubits, avec la deuxième configuration, les grandes erreurs sont plus grandes et les petites erreurs sont plus petites qu'avec la première configuration. Cela correspond à ce que nous attendions, en effet, la première configuration est censée être plus fiable, il est donc logique que les valeurs de ses "outliers" soient plus faibles. La deuxième configuration utilise les mesures de manière plus efficace, il est logique que, lorsqu'elle fonctionne, elle fonctionne mieux. Et si nous avons une idée de la valeur de $\mathbf{M}$ avant d'effectuer la QPT, nous saurons dans quelle partie de la boîte à moustache notre erreur est susceptible de

se situer (avec un peu de chance, pour notre $\mathbf{M}$, ce sera la partie inférieure). Plus il y a de qubits, plus les performances de la deuxième configuration sont mauvaises par rapport à celles de la première. Nous pensons que cela est dû au fait que le conditionnement d'une matrice aléatoire (et $\mathbf{X}$ est essentiellement une matrice aléatoire pour la deuxième configuration) se dégrade lorsque la dimension augmente. Cela rend la QPT plus complexe, quel que soit l'algorithme utilisé.

Il est intéressant de comparer les performances des deux algorithmes (initialisation et maximum de vraisemblance) pour les différentes configurations. Pour deux qubits, on retrouve le facteur 2 (à peu près) entre la médiane de l'erreur post ML et celle l'erreur d'initialisation, ce facteur augmente un peu pour les quantiles d'ordres plus élevés. Ce ratio augmente avec le nombre de qubits, il est clairement inférieur à 2 pour un qubit, et pour 5 qubits avec la première configuration, il est plus proche de 4. On constate aussi que l'algorithme du maximum de vraisemblance peut mal converger et ne pas réduire l'erreur du tout. En effet, pour des erreurs initiales autour de 0,1, l'algorithme de ML semble souvent converger vers un mauvais minimum. Si c'est le cas l'erreur de l'algorithme de maximum de vraisemblance n'a aucune raison d'être plus faible que l'erreur de l'algorithme initial.

Les temps d'exécution médians des algorithmes QST et QPT sont indiqués dans le tableau 5.1. Les simulations ont été codées sur Matlab [3] sur un 210 Intel Xeon silver 4214 2,4-GHz sur un seul "thread". Nous avons séparé la QPT de la QST (malgré le fait que la QST soit indispensable à l'algorithme initial de la QPT) afin de montrer que notre algorithme initial de QPT (calcul de $\widehat{\mathbf{M}}_{LS}$ à partir des résultats de la QST) a un temps d'exécution très faible.

TABLE 5.1 : Temps d'exécution médian de la QST de tous les états et des deux algorithmes de QPT pour la configuration de la figure 3.4 (config. 1) et celle de la figure 3.5 (config. 2).

algo. n_{qb}	1	2	3	4	5
QST config. 1	0,06s	0,14s	0,37s	1,67s	12s
init. QPT config. 1	7e-5s	9e-5s	1e-4s	3e-4s	6e-4s
ML QPT config. 1	0,4s	8s	102s	0,5h	12h
QST config. 2	0,05s	0,10s	0,21s	0,85s	6,9s
init. QPT config. 2	1e-4s	1e-4s	2e-4s	3e-4s	8e-4s
ML QPT config. 2	0,7s	6s	90s	0,35h	6h

Il est clair que le temps d'exécution de l'algorithme d'initialisation de la QPT n'est pas important, il est négligeable devant le temps d'exécution de l'algorithme de QST. Il existe des algorithmes de QST plus rapides dans la littérature, et nous pourrions réduire considérablement le temps d'exécution de la QPT en ne mettant pas en œuvre l'approche basée sur le maximum de vraisemblance (de la section 2.4). Mais nous avons choisi de sacrifier le temps d'exécution au profit de la précision. Pour l'algorithme d'initialisation de la QPT, nous n'avons besoin de calculer que quelques produits scalaires pour la récupération de phase, et quelques produits de matrices $d \times d$ plus une décomposition en valeurs singulières pour résoudre le problème des moindres carrés totaux sous contraintes unitaires, il n'est pas surprenant qu'il soit très rapide. Cela contraste avec le temps d'exécution de l'algorithme de QPT par ML, qui devient très rapidement prohibitif (plusieurs heures pour 5 qubits), alors que nous avons optimisé le code.

5.2 Résultats expérimentaux

Dans la présente section, nous testons expérimentalement notre algorithme de QPT en utilisant un ordinateur quantique à ions piégés sur amazon web services :

[3] https://www.mathworks.com/products/matlab.html

AwsDevice("arn:aws:braket:::device/qpu/ionq/ionQdevice"). Il s'agit d'un circuit quantique programmable, AWS prétend (i) que la fidélité des portes mono-qubit est de 0,9935, ou $\epsilon = 0{,}0806$ avec notre définition de l'erreur (5.1), (ii) que la fidélité est de 0,9602 (ou $\epsilon = 0{,}1995$) pour une porte à deux qubits, (iii) que le temps de cohérence est de $1{,}667s$ et (iv) que la fidélité SPAM (fidélité des états préparés à $|0\rangle$ par rapport aux vecteurs propres des mesures selon Z) vaut $99{,}3 - 99{,}8\%$.

Nous voulons réaliser la QPT sur une porte CNOT à 2 qubits. Comme nous l'avons indiqué dans la section 3.2.3, l'utilisation de deux pas de temps est adaptée à cette porte. Nous avons donc choisi $n_i = d = 4$, $n_s = 2$ avec les quatre états initiaux $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4$ de (3.21) qui sont ceux de la figure 3.4, dans le cas $n_{qb} = 2$. Comme nous l'avons expliqué dans la section 3.2.3, ces états initiaux peuvent être réalisés avec des portes de Hadamard (voir figure 3.4). Nous ne faisons pas confiance à la mise en œuvre des portes de Hadamard et notre algorithme se comportera comme si les états d'entrée étaient totalement inconnus. La valeur cible de la porte CNOT à identifier est :

$$\mathbf{M}_{tg} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5.4}$$

Cette valeur n'est pas non plus utilisée. Le processus à identifier est considéré comme totalement inconnu. La configuration de la figure 5.15 est réalisée. Les $n_t = 2n_{qb} + 1 = 5$ types de

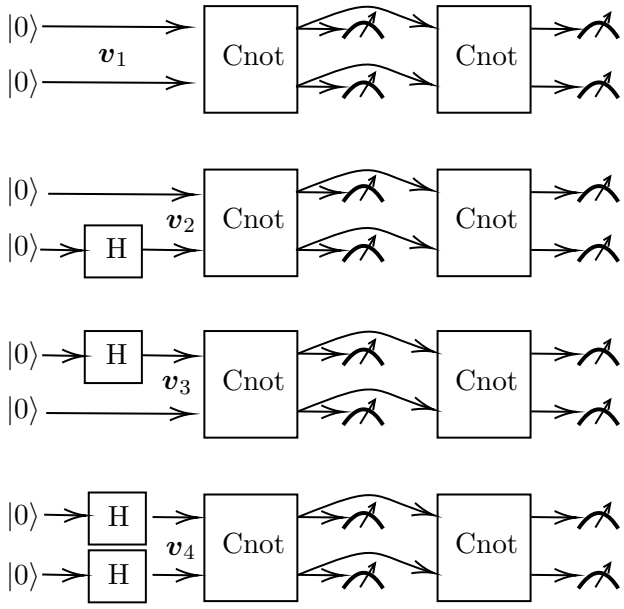


FIGURE 5.15 : Configuration QPT pour une porte CNOT à deux qubits, avec $n_i = 4$ états initiaux et $n_s = 2$ retards temporels. Dans la section 3.1.1, nous avons indiqué que nous appliquons le processus à identifier avant de mesurer un état, et que c'est un choix que nous avons dû faire pour mettre en œuvre la configuration expérimentalement. En effet, le logiciel que nous utilisons ne nous permet pas de mesurer des états qui sont censés avoir été préparés à $|0\rangle$ et qui n'ont jamais été modifiés par une porte quantique.

mesures de la section 2.3 sont chacun effectués $n_c = 250$ fois sur chacun des 8 états mesurés. Au total, $n_c n_t n_s n_i = 250 \times 5 \times 2 \times 4 = 10\,000$ mesures quantiques sont effectuées. Le tableau C.1 (dans l'annexe C.2) indique les nombres de réalisations expérimentales pour chaque résultat

et chaque état. Par exemple, les 243 de la ligne appelée ZZ 00 et de la colonne appelée $\mathbf{M}\boldsymbol{v}_1$ signifient qu'en mesurant $\mathbf{M}\boldsymbol{v}_1$ avec le type de mesure ZZ, nous avons obtenu le premier résultat (appelé 00 ici) 243 fois. À titre de comparaison, le tableau C.2 (également dans l'annexe C.2) donne les espérances (théoriques) des nombres de mesures en supposant que les portes de Hadamard et CNOT font ce qu'elles sont censées faire.

En utilisant les résultats de l'algorithme de QST décrit dans la section 2.3, nous estimons les 8 états mesurés. Nous rencontrons un petit problème ici, les types de mesures définis dans la section 2.3 sont ZZ, ZX, ZY, XX et YX pour deux qubits, mais les mesures que nous avons effectuées sur l'ordinateur quantique sont ZZ, ZX, ZY, XX et YY (le dernier type de mesure n'est pas le même, c'est une erreur de ma part). Il s'avère que l'algorithme de QST de la section 2.3 fonctionne encore avec ces mesures. L'estimation initiale de l'état avec l'algorithme récursif mesuré sera erronée (parce qu'elle ne fonctionne qu'avec le premier ensemble de mesures), mais l'algorithme de QST comporte deux étapes, et il se trouve que la deuxième étape (réglage fin avec le maximum de vraisemblance) corrige l'estimation initiale dans le cas de deux qubits (parce qu'elle peut fonctionner avec les types de mesures qui sont disponibles).

Nous organisons les états estimés dans les matrices $\widehat{\mathbf{X}}$ et $\widehat{\mathbf{Y}}$ définies dans la section 3.1 (ce sont des estimations de $\mathbf{X} = [\mathbf{M}\boldsymbol{v}_1, \mathbf{M}\boldsymbol{v}_2, \mathbf{M}\boldsymbol{v}_3, \mathbf{M}\boldsymbol{v}_4]$ et $\mathbf{Y} = [\mathbf{M}^2\boldsymbol{v}_1, \mathbf{M}^2\boldsymbol{v}_2, \mathbf{M}^2\boldsymbol{v}_3, \mathbf{M}^2\boldsymbol{v}_4]$ respectivement). Voici leurs valeurs numériques :

$$\widehat{\mathbf{X}} = \begin{pmatrix} 1.00 - 0.00i & 0.76 - 0.00i & 0.70 - 0.00i & 0.49 - 0.00i \\ 0.01 - 0.01i & 0.65 - 0.01i & -0.06 + 0.05i & 0.45 - 0.00i \\ -0.02 - 0.03i & 0.02 - 0.03i & 0.06 - 0.03i & 0.54 - 0.06i \\ 0.03 - 0.06i & -0.01 - 0.08i & 0.68 - 0.19i & 0.50 - 0.02i \end{pmatrix}$$

$$\widehat{\mathbf{Y}} = \begin{pmatrix} 1.00 - 0.00i & 0.72 - 0.00i & 0.70 - 0.00i & 0.54 - 0.00i \\ 0.01 + 0.01i & 0.70 + 0.06i & -0.02 - 0.00i & 0.46 + 0.04i \\ 0.03 - 0.00i & -0.02 - 0.01i & 0.72 - 0.02i & 0.51 + 0.08i \\ 0.00 - 0.01i & 0.01 - 0.01i & 0.01 - 0.02i & 0.47 + 0.08i \end{pmatrix}$$

Nous utilisons ensuite l'algorithme de récupération de phase de la section 3.1.3 pour modifier la phase des colonnes de $\widehat{\mathbf{Y}}$.

Nous utilisons ensuite la méthode de la section 3.1.2 pour trouver la matrice unitaire $\widehat{\mathbf{M}}_{LS}$ qui lie le mieux $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ (la version rephasée de $\widehat{\mathbf{Y}}$). Nous modifions ensuite sa phase globale afin de la comparer à $\mathbf{M}_{tg}$:

$\widehat{\mathbf{M}}_{LS} \longleftarrow e^{i\theta}\widehat{\mathbf{M}}_{LS}$ avec $\theta = arg\left(tr(\mathbf{M}_{tg}\widehat{\mathbf{M}}^*_{LS})\right)$.

Nous appliquons ensuite l'algorithme de maximum de vraisemblance de la section 3.3 pour calculer $\widehat{\mathbf{M}}_{ML}$, puis nous modifions sa phase globale de la même façon que nous l'avons fait pour $\widehat{\mathbf{M}}_{LS}$.

Les résultats (arrondis) sont les suivants :

$$\widehat{\mathbf{M}}_{LS} = \begin{pmatrix} 0.982 - 0.171i & -0.023 - 0.020i & 0.015 + 0.022i & 0.007 + 0.067i \\ 0.016 - 0.023i & 0.995 - 0.086i & 0.012 + 0.032i & 0.026 + 0.008i \\ -0.001 + 0.070i & -0.024 + 0.009i & 0.076 - 0.021i & 0.991 + 0.077i \\ -0.013 + 0.018i & -0.012 + 0.031i & 0.978 + 0.180i & -0.070 - 0.041i \end{pmatrix} \tag{5.5}$$

$$\widehat{\mathbf{M}}_{ML} = \begin{pmatrix} 0.973 - 0.212i & 0.013 + 0.0018i & 0.033 + 0.064i & -0.002 - 0.046i \\ -0.006 + 0.006i & 0.993 - 0.090i & 0.011 - 0.075i & 0.031 + 0.015i \\ 0.001 - 0.038i & -0.038 + 0.008i & 0.095 - 0.069i & 0.985 + 0.113i \\ -0.035 + 0.068i & -0.000 - 0.073i & 0.969 + 0.189i & -0.066 - 0.095i \end{pmatrix} \tag{5.6}$$

Les modules sont proches de leurs valeurs cibles mais il y a des erreurs assez importantes sur les phases qui ne peuvent pas être corrigées par un déphasage global. Ceci est particulièrement visible entre la première colonne et la troisième colonne. La distance entre $\widehat{\mathbf{M}}_{LS}$ et la cible $\mathbf{M}_{tg}$ peut être définie comme $\mu_p(\widehat{\mathbf{M}}_{LS}, \mathbf{M}_{tg}) \simeq 0{,}11$ avec μ_p défini dans (5.1). D'après la figure 5.1, 0,11 est une erreur très raisonnable avec $n_c = 250$ dans la configuration $n_i = 4$, $n_s = 2$.

L'estimée du maximum de vraisemblance nous donne aussi des estimées des états initiaux $\widehat{\boldsymbol{v}}_1, ..., \widehat{\boldsymbol{v}}_4$, à comparer avec les états cibles de (3.21). Nous avons changé la phase globale de chaque état $\widehat{\boldsymbol{v}}_1, ..., \widehat{\boldsymbol{v}}_4$ pour qu'ils soient les plus proches possible des états de (3.21) :

$$\widehat{\boldsymbol{v}}_1 = \begin{pmatrix} 0.997 + 0.00i \\ 0.03 - 0.01i \\ 0.06 - 0.05i \\ 0.00 - 0.00i \end{pmatrix} \widehat{\boldsymbol{v}}_2 = \begin{pmatrix} 0.72 + 0.04i \\ 0.69 - 0.04i \\ 0.04 - 0.01i \\ 0.04 - 0.01i \end{pmatrix} \widehat{\boldsymbol{v}}_3 = \begin{pmatrix} 0.63 + 0.23i \\ -0.03 + 0.05i \\ 0.70 - 0.23i \\ 0.01 + 0.05i \end{pmatrix} \widehat{\boldsymbol{v}}_4 = \begin{pmatrix} 0.48 + 0.07i \\ 0.43 + 0.07i \\ 0.55 - 0.08i \\ 0.50 - 0.06i \end{pmatrix} \quad (5.7)$$

Si nous nous fions à ces estimations, l'erreur la plus importante se situe entre $\widehat{\boldsymbol{v}}_3$ et $\boldsymbol{v}_3^{tg}$ de (3.21). La différence de phase entre la première composante et la troisième composante est très importante, la porte de Hadamard utilisée sur le premier qubit pourrait être non-conforme au modèle (dans la base de référence définie par les mesures). La configuration semi-aveugle fait que nos estimées de $\mathbf{M}$ sont résistantes à ce type d'erreurs. Si les états initiaux étaient exactement ceux de (5.7) et si la matrice associée à la porte CNOT était celle de (5.5), les espérances des mesures seraient celles du tableau C.3 de l'annexe C.2.

Nous pouvons aussi calculer la borne de Cramér-Rao et en déduire la covariance des parties réelles et parties imaginaires des éléments de $\widehat{\mathbf{M}}_{ML}$. Cela donne une matrice de covariance de taille 32×32, les valeurs de ses éléments sont représentées sur la figure 5.16

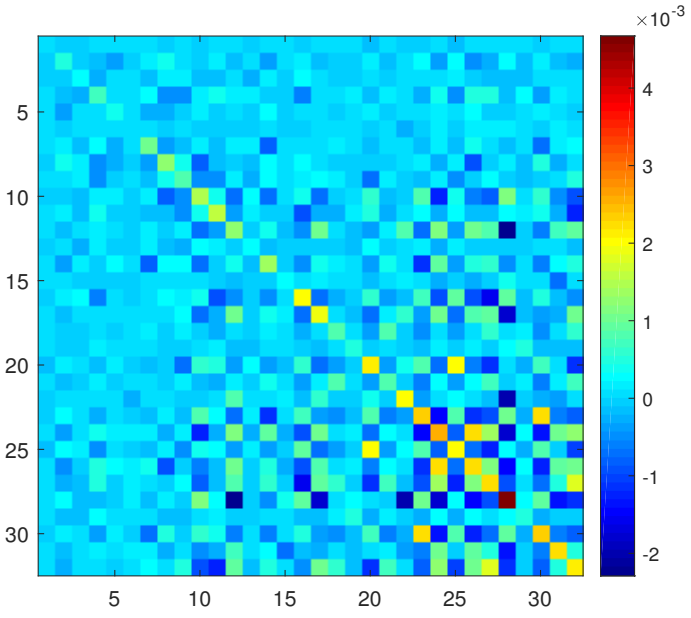


FIGURE 5.16 : Valeurs des coefficients de la matrice de covariance 32×32 des parties réelles et imaginaires des coefficients de $\widehat{\mathbf{M}}_{ML}$. Indice des lignes en abscisse, indice des colonnes en ordonnée, les couleurs correspondent aux valeurs des éléments.

Si on n'est pas intéressé par les covariances des erreurs entre les différents coefficients, on peut se limiter aux éléments de la diagonale de la matrice de covariance. On obtient un vecteur de taille 32 dont on prend la racine carrée (coefficient par coefficient) pour calculer les écarts

types, et on transforme le vecteur de taille 32 en deux matrices 4×4 : Σ_r et Σ_i qui contiennent les écarts types de parties réelles et imaginaires respectivement des coefficients de $\widehat{\mathbf{M}}_{ML}$:

$$\Sigma_r = \begin{pmatrix} 0.0034 & 0.0203 & 0.0276 & 0.0178 \\ 0.0223 & 0.0070 & 0.0388 & 0.0365 \\ 0.0167 & 0.0315 & 0.0395 & 0.0161 \\ 0.0267 & 0.0358 & 0.0347 & 0.0447 \end{pmatrix} \Sigma_i = \begin{pmatrix} 0.0431 & 0.0286 & 0.0442 & 0.0222 \\ 0.0281 & 0.0450 & 0.0468 & 0.0478 \\ 0.0218 & 0.0485 & 0.0470 & 0.0469 \\ 0.0463 & 0.0508 & 0.0683 & 0.0466 \end{pmatrix}. \tag{5.8}$$

Ces valeurs ne sont intéressantes que si on fait l'hypothèse que le modèle de vraisemblance (gaussien régularisé) modélise bien les imperfections du circuit. Si on admet que la vraisemblance est la bonne, on peut rejeter l'hypothèse que la vraie matrice $\mathbf{M}$ est $\mathbf{M}_{tg}$, car $\widehat{\mathbf{M}}_{ML}$ est trop différente de $\mathbf{M}_{tg}$, par rapport aux écarts types de (5.8). Même si on ne regarde que la partie imaginaire du coefficient en première ligne première colonne, la différence est de 0,212, et l'écart type de la partie imaginaire du premier élément de $\widehat{\mathbf{M}}_{ML}$ est presque 5 fois plus petit (0,0454). Pour une loi gaussienne[4], il est presque impossible d'avoir une observation à 5σ.

5.3 Conclusion

Les simulations que nous avons réalisées dans le cadre de la figure 3.4 (pour $n_{qb} = 2$, $n_i = 4, n_s = 2$) nous ont permis de constater que (i) l'erreur de QPT varie de façon linéaire avec la racine de l'inverse du nombre de copies préparées, (ii) nos algorithmes sont presque insensibles aux erreurs systématiques, (iii) nous sommes plus résistants aux erreurs de QST que l'algorithme de [BKD14], et (iv) si le modèle de la vraisemblance modélise correctement les mesures, alors notre estimée de l'erreur avec la borne de Cramér-Rao est bonne.

Nous avons aussi validé notre algorithme de QMT avec des simulations. Comme on pouvait s'y attendre, il n'améliore la précision que s'il y a une erreur (non-nulle mais pas non plus trop importante) sur le modèle des mesures.

Au delà de la configuration "principale" avec $n_{qb} = 2$, $n_i = 4$ et $n_s = 2$, nous avons fait varier le nombre de qubit, et comparé la configuration de la figure 3.4 ($n_i = d, n_s = 2$) avec celle de la figure 3.5 ($n_i = 1, n_s = d + 1$). La conclusion est que cette dernière configuration peut donner une erreur plus grande ou plus petite en fonction de la valeur du processus à identifier, et qu'elle perd sont intérêt pour $n_{qb} > 3$.

Finalement, nous avons testé nos algorithmes de QPT sur des données expérimentales. Les résultats sont cohérents avec les spécifications du "hardware".

[4] L'erreur sur $\widehat{\mathbf{M}}_{ML}$ est censée être gaussienne car le modèle de la vraisemblance des erreurs de mesures est gaussien avec un écart type assez faible pour que l'estimateur soit considéré comme linéaire dans le voisinage des mesures. L'estimateur va donc transformer une erreur gaussienne sur les mesures en erreur gaussienne sur les coefficients de $\widehat{\mathbf{M}}_{ML}$.

Conclusion et perspectives

Les travaux de recherche de cette thèse visaient à élaborer des algorithmes de tomographie de processus quantiques (QPT) adaptés aux processus unitaires, résistants aux erreurs systématiques, et qui fonctionnent avec des mesures simples.

— Nous avons choisi de faire l'hypothèse que les états considérés sont purs. Ce n'est pas particulièrement irréaliste, dans un système fermé avec une évolution unitaire, tout état que l'on peut générer plusieurs fois de façon identique est pur. Les états purs ont beaucoup moins de paramètres que les états mélanges, cette hypothèse réduit donc le nombre de mesures nécessaires pour identifier les états. Dans le chapitre 3, nous avons défini un algorithme de QPT qui, comme beaucoup d'autres, se base sur la tomographie d'états quantiques (QST) des états mesurés. Comme la QST d'états purs est assez peu développée dans la littérature, nous avons dû développer des algorithmes de QST originaux dans le Chapitre 2.

— Nous ne sommes pas les premiers à chercher à estimer un processus unitaire à partir des états d'entrée et de sortie. Dans [RGK13], Reich et al. ont donné une condition nécessaire et suffisante sur les états d'entrée pour que le processus unitaire soit identifiable. Dans [BKD14], Baldwin et al. ont choisi des états (intriqués) qui respectent la condition d'identifiabilité de [RGK13] et avec lesquels un algorithme particulièrement simple permet d'identifier le processus. Dans le Chapitre 3, nous avons introduit un algorithme original de QPT qui fonctionne avec n'importe quels états purs qui vérifient la condition de [RGK13] en entrée. Nous avons aussi montré que notre algorithme de base est rapide (au point que son temps d'exécution est négligeable devant le temps d'exécution de notre algorithme de QST), et que ses performances sont meilleures que celle de l'algorithme de [BKD14]. Nous avons conclu le Chapitre 3 avec un algorithme de maximum de vraisemblance plus précis, mais qui a besoin d'un bon point d'initialisation. Cet algorithme est plus précis et plus rapide si on suppose que les états initiaux sont non intriqués, nous avons donc fait cette hypothèse (qui a aussi l'avantage d'être attrayante car les états non-intriqués sont plus faciles à préparer de façon précise).

— Avant le Chapitre 4, nos hypothèses pouvaient être critiquées en remarquant que nous choisissions de faire confiance aux mesures non intriquées, alors que nous considérions que les portes mono-qubits que nous utilisons pour préparer les états initiaux non intriqués n'étaient pas fiables (les états initiaux sont estimés à partir des mesures), or, pour certaines architectures, les mesures non intriquées sont effectuées avec des portes mono-qubit. Dans le Chapitre 4, nous avons donc élaboré un algorithme de tomographie de mesures quantiques (QMT) qui estime les paramètres des mesures et réalise la QPT simultanément.

Nous pensons que les hypothèses que nous avons faites sont cohérentes et pertinentes. La Fig. 5.17 récapitule toutes les hypothèses qui peuvent être faites lors de la QPT, sur les états d'entrée, le processus inconnu à identifier et les mesures effectuées.

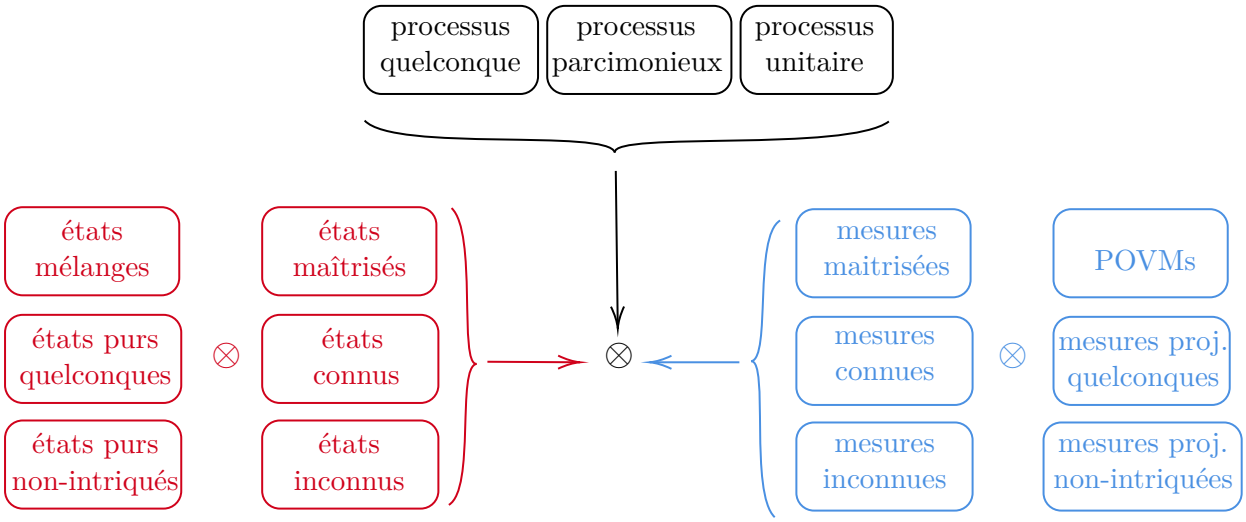


FIGURE 5.17 : Hypothèses pour les algorithmes de QPT, sur les états d'entrée en rouge, le processus inconnu à identifier en noir et les mesures effectuées en bleu ("mesures proj." signifie "mesures projectives"). Les produits tensoriels ($\otimes$) signifient que l'on peut envisager n'importe quelle combinaison des propriétés. Par exemple, à gauche, les états peuvent être des états mélanges maitrisés, mélanges connus, mélanges inconnus, purs maitrisés, ..., purs, non-intriqués et inconnus (neuf possibilités). Par "maîtrisé" nous voulons dire "connu et dont la valeur est imposée par la méthode".

Tous les algorithmes de QPT se basent sur une des hypothèses de la Fig 5.17. Dans le cas le plus extrême, la GST est adaptée à des états mélanges inconnus, un processus quelconque à identifier (en pratique, il y a plusieurs processus) et un type de mesure projective intriqué. Le premier algorithme de QPT de [CN97] utilise des états mixtes (ils peuvent être purs quelconques ou purs intriqués, mais ce n'est pas nécessaire) connus, et des mesures projectives non-intriquées maitrisées (connues et imposées par la méthode) pour identifier un processus quelconque. À l'autre extrême, le premier algorithme de [BKD14] identifie un processus unitaire avec des états purs maitrisés et un POVM maitrisé.

Nos contributions principales sont (i) l'algorithme de QST sans initialisation de la section 2.3, (ii) l'algorithme de QST sans initialisation de la section 2.2 (nous ne le considérons pas vraiment comme une contribution originale, mais il est important), (iii) l'algorithme de QST par maximisation de la vraisemblance de la section 2.4, (iv) l'algorithme initial de QPT de la section 3.1 (contribution principale), (v) l'algorithme de QPT par maximisation de la vraisemblance (autre contribution majeure), (vi) l'algorithme de QMT (qui peut se faire en parallèle de la QPT sur le même dispositif) du Chapitre 4, cet algorithme a des hypothèses très particulières adaptées aux configurations de QPT que nous considérons.

Récapitulons les hypothèses que nous avons faites (dans nos contributions) sur les états, le processus et les mesures :

- Nos algorithmes de QPT ont été pensés pour des états d'entrée non intriqués, mais :
  - Tous les algorithmes de QST ((i), (ii) et (iii)) sont pensés pour des états intriqués, car les états mesurés (que l'on identifie avec la QST) sont, en général, intriqués par le processus à identifier.
  - L'algorithme de QPT (iv) est adapté aux états purs non intriqués et (v) est adapté aux états intriqués, mais peut se généraliser à l'ensemble des états purs au prix d'une perte de précision (voir section 5.1.4)

  - En utilisant des algorithmes de QST d'états mixtes de la littérature, nous pouvons adapter nos algorithmes de QPT (iv) et (v) pour identifier le processus (il suffit de raisonner sur les vecteurs propres des matrices densité) en supposant que les valeurs propres non-nulles des matrices densité des états d'entrée sont distinctes. Nous avons choisi de ne pas développer cette idée, car les ordinateurs quantiques fonctionnent avec des états purs. En pratique les états seront presque purs avec une grande valeur propre et $d-1$ valeurs propres beaucoup plus faibles. Utiliser les vecteurs propres associés à des faibles valeurs propres complique l'algorithme pour peu de gain car ces vecteurs propres sont mal estimés.
  - Nous avons supposé que les états de la figure 5.17 sont inconnus, mais, bien entendu, s'ils sont connus (ou maitrisés), nos algorithmes fonctionnent. On peut même utiliser l'information sur les états connus et changer n_s en $n_s - 1$ (voir section 5.1.3).

- Nous supposons que le processus identifié est unitaire, en pratique, ce ne sera jamais le cas, mais plus le processus sera proche d'un processus unitaire, mieux nos algorithmes vont fonctionner, voir section 5.1.5. Le choix de cette hypothèse est justifié dans la section 1.12.1.

- Les algorithmes de QST et QPT ont été pensés pour des mesures non intriquées et maitrisées (les produits tensoriels des mesures X, Y et Z de section 2.1.1). Cependant :

  - Tous nos algorithmes de QPT et QST sauf (i) peuvent être adaptés pour des mesures connues, donc si les mesures ne sont pas exactement conformes au modèle de la section 2.1.1, nos algorithmes peuvent fonctionner tant que les mesures sont connues. (i) ne peut pas être adapté, mais il n'est utilisé que pour l'initialisation de la QST (avant d'être raffiné par (ii)), la QST peut donc fonctionner normalement tant que l'estimée donnée par (i) n'est pas trop erronée, ce sera le cas si les mesures ne sont pas trop différentes du modèle de la section 2.1.1.
  - L'algorithme de QMT (vi) nous permet de réaliser la QPT avec des mesures non intriquées inconnues.

Nous pensons que ces hypothèses sont réalistes, elles se résument à (i) on sait préparer le même état plusieurs fois (sans nécessairement connaître sa valeur) (ii) on sait faire des opérations sur un ou plusieurs qubits sans interagir avec l'environnement.

Avec plus de temps, nous aurions pu essayer d'étendre nos algorithmes à des modèles plus généraux, même si on perd l'hypothèse que le système est fermé (e.g. avec des "crosstalk errors" sur les mesures ou la décohérence des états mesurés). Pour ce faire, nous avons deux options (i) nous pouvons ajouter de nouveaux paramètres, ou, (ii) nous pouvons intégrer ces erreurs dans le modèle de vraisemblance et garder le même modèle. A moins que le modèle plus général n'ajoute pas trop de paramètres (ce qui est rarement le cas pour les systèmes ouverts), nous préférons la piste (ii). En effet, le modèle de vraisemblance que nous utilisons (le modèle Gaussien régularisé de la section 2.4.3) ne prend en compte que l'erreur "multinomiale" (due au caractère fini du nombre de mesures) et une "erreur fourre-tout" gaussienne (issue de la régularisation des probabilités). Si nous savons que le processus introduit une décohérence par exemple, alors, nous pouvons changer la régularisation dans le modèle gaussien pour que l'incertitude sur les mesures effectuées sur les états sur lesquels $\mathbf{M}$ a été appliqué plusieurs fois pour que ces états soient considérés comme moins digne de confiance.

Une autre idée pour exploiter le fait que nos algorithmes utilisent le maximum de vraisemblance est d'utiliser la valeur de la vraisemblance au maximum trouvé pour estimer une p-value, cela permettrait de donner un indice de confiance au $\widehat{\mathbf{M}}_{ML}$ trouvé. Nous sommes capable de calculer cette p-value, mais nous n'avons pas donné ces résultats dans nos tests pour deux raisons :

(i) la p-value perd tout son sens si on n'est pas sûr du modèle de vraisemblance, c'est moins vrai pour le maximum de vraisemblance[5] (ii) en simulation, nos p-values sont toutes très proches de 1 car l'estimateur peut jouer sur tous les paramètres pour rendre l'erreur plus vraisemblable. Cependant, si nous étions plus sûrs du modèle de vraisemblance, ces objections disparaîtraient.

Nous pourrions aussi améliorer notre estimée initiale de $\mathbf{M}$: $\widehat{\mathbf{M}}_{LS}$ avec un algorithme plus léger que l'algorithme de maximisation de la vraisemblance. $\widehat{\mathbf{M}}_{LS}$ est définie comme la matrice unitaire qui transforme $\widehat{\mathbf{X}} + \mathbf{\Delta}_X$ en $\widetilde{\mathbf{Y}} + \mathbf{\Delta}_Y$, où $\mathbf{\Delta}_X$ et $\mathbf{\Delta}_Y$ sont les solutions de (3.11). (3.11) minimise $||\mathbf{\Delta}_X||^2 + ||\mathbf{\Delta}_Y||^2$, avec $\mathbf{\Delta}_X$ et $\mathbf{\Delta}_Y$ tels qu'il existe une matrice unitaire qui transforme $\widehat{\mathbf{X}} + \mathbf{\Delta}_X$ en $\widetilde{\mathbf{Y}} + \mathbf{\Delta}_Y$ (c'est le problème de Procrustes orthogonal). Comme nous l'avons remarqué dans la section 3.1.2, $\mathbf{\Delta}_X$ et $\mathbf{\Delta}_Y$ peuvent être compris comme l'erreur de $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ définis respectivement dans les équations (3.4) et (3.7) par rapports aux quantités qu'ils sont censés estimer : $\mathbf{X}$, défini dans (3.4) (et avec toutes les colonnes re-phasées pour que le premier élément soit un réel positif (comme $\widehat{\mathbf{X}}$)) et $\mathbf{MX}$. Cette approche (minimiser $||\mathbf{\Delta}_X||^2 + ||\mathbf{\Delta}_Y||^2$) n'est optimale d'un point de vue maximum de vraisemblance que si l'erreur est gaussienne centrée iid sur chaque élément de $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$. L'hypothèse que l'erreur est gaussienne n'est pas aberrante, l'erreur sur les mesures est multinomiale, donc asymptotiquement gaussienne, et même sur les traitements que l'on effectue sur les mesures ne sont pas linéaires, si la variance de l'erreur est assez faible, on peut les linéariser[6]. On devrait aussi pouvoir calculer les matrices de covariance des erreurs sur $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$, et on peut adapter la solution du problème de Procrustes avec une erreur Gaussienne colorée (voir [Vik06]) pour avoir une meilleure solution que $\widehat{\mathbf{M}}_{LS}$ qui prend en compte la covariance de l'erreur. Nous n'avons pas exploité cette idée car, pour une première estimée fiable mais peu précise de $\mathbf{M}$, avoir une estimée qui ne dépend pas du modèle de vraisemblance des mesures (en lequel nous n'avons pas une foi absolue) nous semble cohérent.

[5] l'estimateur des moindres carrés, par exemple, correspond au maximum de vraisemblance si et seulement si l'erreur est gaussienne centrée iid sur toutes les composantes, mais il est utilisé pour toutes les erreurs centrées, même si on ne connaît pas la distribution

[6] Dans la section 5.1.7, on voit que les bornes de Cramér-Rao sont assez proches de la vraie erreur, et elles sont issues d'une linéarisation de la fonction qui associe $\widehat{\mathbf{M}}_{ML}$ aux mesures. Il est raisonnable de penser que les fonctions qui associent $\widehat{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ aux mesures sont aussi linéarisables.

Liste des publications

- Juillet 2021 : article de conférence

  F. Verdeil, Y. Deville et A. Deville.

  “Two-qubit unitary quantum process tomography by multiple-delay output measurements for one unknown input pure state value”

  Dans *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 161–165. IEEE, Rio de Janeiro, Brazil, 2021.

- Août 2022 : Article de conférence pour MaxEnt 2022 publié dans MDPI

  F. Verdeil et Y. Deville.

  “Two Unitary Quantum Process Tomography Algorithms Robust to Systematic Errors"

  Dans Physical Sciences Forum. MDPI, 2022. p. 29

- Janvier 2023 : Article de journal

  F. Verdeil et Y. Deville.

  “Pure-state tomography with parallel unentangled measurements”

  Dans Physical Review A vol. 107, pp. 012408.

- Décembre 2023 article de journal publié dans PRA

  F. Verdeil et Y. Deville

  “Unitary quantum process tomography with unreliable pure input states”

  Dans Physical Review A vol. 108, pp. 062410.

Annexes

Annexe A

Annexe du chapitre 2

Dans cette annexe, nous donnons les démonstrations de résultats du Chapitre 2.

A.1 Paramétrisation des matrices unitaires de taille 2

Soit $\mathbf{P}$ une matrice unitaire de taille 2. Par définition $\mathbf{P}\mathbf{P}^* = \mathbf{P}^*\mathbf{P} = \mathbf{I}_2$, montrons

$\exists \theta, \phi_1, \phi_2, \phi_3$ t.q. $\begin{pmatrix} \cos(\theta)e^{i\phi_1} & -\sin(\theta)e^{i\phi_2} \\ \cos(\theta)e^{i\phi_3} & \cos(\theta)e^{i(\phi_2+\phi_3-\phi_1)} \end{pmatrix}$.

Définissons les lignes et les colonnes de $\mathbf{P}$: $\mathbf{P} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \end{bmatrix}$. On a $\begin{bmatrix} \boldsymbol{c}_1^* \\ \boldsymbol{c}_2^* \end{bmatrix} \times \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ et $\begin{bmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{l}_1^* & \boldsymbol{l}_2^* \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, donc les vecteurs lignes et colonnes sont de norme 1, commençons par étudier le $|\boldsymbol{c}_1|$ et $|\boldsymbol{c}_2|$ (les modules terme à terme des colonnes). Ce sont des vecteurs normés de deux réels positifs 1, il existe deux angles θ_1 et θ_2 uniques entre 0 et $\pi/2$ tels que $\begin{bmatrix} |\boldsymbol{c}_1| & |\boldsymbol{c}_2| \end{bmatrix} = \begin{pmatrix} \cos(\theta_1) & \cos(\theta_2) \\ \sin(\theta_1) & \sin(\theta_2) \end{pmatrix}$. Les vecteurs $|\boldsymbol{l}_1|$ et $|\boldsymbol{l}_2|$ sont aussi des vecteurs normés de deux réels positifs, il existe donc deux angles θ_3 et θ_4 uniques entre 0 et $\pi/2$ tels que $\begin{bmatrix} |\boldsymbol{l}_1| \\ |\boldsymbol{l}_2| \end{bmatrix} = \begin{pmatrix} \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix}$. Donc $\begin{pmatrix} \cos(\theta_3) & \sin(\theta_3) \\ \cos(\theta_4) & \sin(\theta_4) \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & \cos(\theta_2) \\ \sin(\theta_1) & \sin(\theta_2) \end{pmatrix}$, or, sur $[0, \pi/2]$, les fonctions cosinus et sinus sont injectives, on a donc $\theta_1 = \theta_3 = \pi/2 - \theta_4 = \pi/2 - \theta_2$. On a donc $|P| = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$ (on a juste renommé θ_1 en θ), il existe donc des phases $\phi_1, \phi_2, \phi_3, \phi_4$ telles que $|P| = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$, sans le module, $\mathbf{P}$ s'écrit donc $\begin{pmatrix} \cos(\theta)e^{i\phi_1} & -\sin(\theta)e^{i\phi_2} \\ \cos(\theta)e^{i\phi_3} & \cos(\theta)e^{i(\phi_4)} \end{pmatrix}$, le moins en première ligne deuxième colonne a juste pour effet de changer ϕ_2 en $\phi_2 - \pi$. Il ne nous reste plus qu'à montrer que $\phi_4 = \phi_2 + \phi_3 - \phi_1[2\pi]$, on peut utiliser $\boldsymbol{l}_1\boldsymbol{l}_2^* = 0 \Rightarrow \cos(\theta)\sin(\theta)e^{i(\phi_1-\phi_3)} - \cos(\theta)\sin(\theta)e^{i(\phi_2-\phi_4)} = 0 \Rightarrow \cos(\theta)\sin(\theta)(1 - e^{i(\phi_2-\phi_4-\phi_1+\phi_3))}) = 0$ donc soit (i) $\cos(\theta) = 0$, et alors on peut changer ϕ_4 en $\phi_2 + \phi_3 - \phi_1$ sans changer la valeur de la matrice, soit (ii) $\sin(\theta)$ est nul et alors on peut changer ϕ_2 en $\phi_1 - \phi_3 + \phi_4$ sans changer la valeur de la matrice pour que $\phi_4 = \phi_2 + \phi_3 - \phi_1$, soit (iii) $(1 - e^{i(\phi_2-\phi_4-\phi_1+\phi_3))}) = 0$ est nul, alors $\phi_4 = \phi_2 + \phi_3 - \phi_1[2\pi]$ cqfd. Nous avons donc montré que toute matrice unitaire de taille 2 s'écrit $\begin{pmatrix} \cos(\theta)e^{i\phi_1} & -\sin(\theta)e^{i\phi_2} \\ \cos(\theta)e^{i\phi_3} & \cos(\theta)e^{i(\phi_2+\phi_3-\phi_1)} \end{pmatrix}$. Réciproquement, il est trivial de vérifier que ces matrices sont unitaires.

A.2 Vecteurs propres de mesures séparables

Soit une mesure $\mathcal{M}$ à $d = 2^{n_{qb}}$ résultats possibles sur un système à n_{qb} qubits. On considère que la mesure est non intriquée, cela veut dire que les résultats possibles peuvent être renommés en $0,..0$, $0,..01$, ..., $1...1$ de façon à ce qu'il existe n_{qb} mesures mono-qubit à 2 résultats possibles 0 et 1, $\mathcal{M}_1, ..., \mathcal{M}_{n_{qb}}$ telles que, pour tout état multi-qubit représenté par $\boldsymbol{v}$, le résultat de $\mathcal{M}$ sur $\boldsymbol{v}$ est la concaténation des mesures $\mathcal{M}_1$ appliquée au premier qubit, ..., $\mathcal{M}_{n_{qb}}$ appliquée au dernier qubit.

Nous voulons étudier la matrice de vecteurs propres de $\mathcal{M}$, c'est-à-dire la matrice dont les lignes sont les transconjuguées des vecteurs $\boldsymbol{v}_{0...0}, ..., \boldsymbol{v}_{1...1}$ qui, quand mesurés par $\mathcal{M}$ donnent $0,..0$, ..., $1,..1$ avec probabilité 1. Comme $\mathcal{M}$ est une mesure à d résultats possibles, ces vecteurs propres sont uniques et si on arrive à en exhiber, on les aura identifiés. Or, si on définit $\boldsymbol{v}_0^k$, $\boldsymbol{v}_1^k$ comme les vecteurs propres de $\mathcal{M}_k \forall k \in \{1, ..., n_{qb}\}$ (transconjuguées des deux lignes de la matrice des vecteurs propres), par définition, ils représentent les états mono-qubits qui valent respectivement 0 et 1 avec probabilité 1 ; quand ils sont mesurés avec le type de mesure $\mathcal{M}_k$, alors l'état représenté par $\boldsymbol{v}_0^1 \otimes ... \otimes \boldsymbol{v}_0^{n_{qb}}$ vaut $0,..0$ avec probabilité 1 quand il est mesuré est $\mathcal{M}$ (car les qubits sont dans des états purs définis comme les vecteurs propres des mesures mono-qubits). On peut exhiber tous les vecteurs propres de $\mathcal{M}$ de la même façon : l'état représenté par $\boldsymbol{v}_0^1 \otimes ... \otimes \boldsymbol{v}_0^{n_{qb}} \otimes \boldsymbol{v}_1^{n_{qb}}$ vaut $0,..01$ avec probabilité 1 quand il est mesuré est $\mathcal{M}$, ..., l'état représenté par $\boldsymbol{v}_1^1 \otimes ... \otimes \boldsymbol{v}_1^{n_{qb}}$ vaut $1,..1$ avec probabilité 1 quand il est mesuré avec le type de mesure $\mathcal{M}$. La matrice des vecteurs propres de $\mathcal{M}$ est donc

$$\mathbf{P}_{\mathcal{M}} = \begin{bmatrix} (\boldsymbol{v}_0^1 & \otimes ... \otimes & \boldsymbol{v}_0^{n_{qb}})^* \\ \vdots & & \vdots \\ (\boldsymbol{v}_1^1 & \otimes ... \otimes & \boldsymbol{v}_1^{n_{qb}})^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_0^{1*} \\ \boldsymbol{v}_1^{1*} \end{bmatrix} \otimes ... \otimes \begin{bmatrix} \boldsymbol{v}_0^{n_{qb}*} \\ \boldsymbol{v}_1^{n_{qb}*} \end{bmatrix} = \mathbf{P}_1 \otimes ... \otimes \mathbf{P}_{n_{qb}} \tag{A.1}$$

où les $\mathbf{P}_k$ sont les matrices de vecteurs propres des $\mathcal{M}_k$, c'est-à-dire $\begin{bmatrix} \boldsymbol{v}_0^{1*} \\ \boldsymbol{v}_1^{1*} \end{bmatrix}$.

A.2.1 Matrice de covariance

La présente annexe vise à calculer la loi asymptotique de $\sqrt{n_c}\boldsymbol{\varepsilon} = \sqrt{N}(\widehat{\boldsymbol{p}} - \boldsymbol{p})$ définie dans la section 2.4.3 et à simplifier l'expression de la vraisemblance de $\boldsymbol{\varepsilon}$. Nous considérons que $\boldsymbol{p}$ contient les probabilités d'un seul type de mesure à d résultats possibles sur un seul état. La généralisation à plusieurs types de mesures est simple car les erreurs sur les différentes mesures sont indépendantes (voir la section A.2.3). Le seul vecteur aléatoire dans $\boldsymbol{\varepsilon}$ est $\widehat{\boldsymbol{p}}$ défini comme le vecteur qui contient les probabilités d'échantillonnage de chacun des d résultats. Donc $\widehat{\boldsymbol{p}} = \frac{1}{n_c}\boldsymbol{n}$ où chaque composante n_i de $\boldsymbol{n}$ contient le nombre de fois où le i-ème résultat a été observé. Par définition, $\boldsymbol{n}$ suit une distribution multinomiale caractérisée par le nombre de lancers n_c et les probabilités théoriques de chaque résultat contenues dans $\boldsymbol{p}$. L'espérance et la matrice de covariance de la distribution multinomiale sont connues : $E(\boldsymbol{n}) = n_c\boldsymbol{p}$ et $Cov(\boldsymbol{n}) = n_c(diag(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T)$.

Nous voulons utiliser le théorème de la limite centrale, nous écrivons donc $\boldsymbol{n}$ comme une somme : $\boldsymbol{n} = \sum_{k=1}^{n_c} \boldsymbol{\delta}_k$ où les $\{\boldsymbol{\delta}_k\}_k$ sont iid, tels que $\boldsymbol{\delta}_k$ contient $d-1$ zéros et un 1 à un indice aléatoire $i_k \in \{1, ..., n_c\}$ dont la fonction de densité est $j \Rightarrow p_j$ (c'est-à-dire que la probabilité que i_k prenne la valeur $j \in \{1, ..., n_c\}$ est p_j, le j -ème élément de $\boldsymbol{p}$). $\boldsymbol{\delta}_k$ suit une distribution multinomiale avec $n_c = 1$ lancers. Son espérance est donc $\boldsymbol{p}$ et sa matrice de covariance est $diag(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T$. Par conséquent, $\boldsymbol{\varepsilon}$ est la différence entre la moyenne empirique de $\boldsymbol{\delta}_k$ avec n_c réalisations et son espérance. D'après le théorème de la limite centrale, lorsque $n_c \to +\infty$, la

distribution de $\sqrt{n_c}\boldsymbol{\varepsilon}$ tend vers une distribution normale multivariée centrée, et sa matrice de covariance est $\boldsymbol{\Sigma}_{\mathbf{full}} = diag(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T$.

A.2.2 Vraisemblance

La manière la plus simple de calculer la vraisemblance d'un vecteur qui suit une distribution normale multivariée consiste à inverser la matrice de covariance. Si la matrice de covariance n'est pas inversible, alors elle n'est pas de plein rang, ce qui signifie qu'au moins un élément (aléatoire) du vecteur aléatoire dépend linéairement (avec une décomposition linéaire déterministe) des autres et que ce élément n'est donc pas nécessaire pour calculer la vraisemblance. Ces éléments peuvent être supprimées et la vraisemblance du vecteur plus petit est la même que la vraisemblance du vecteur original. Dans notre cas, la somme des composantes de $\sqrt{n_c}\boldsymbol{\varepsilon}$ est égale à zéro, sa matrice de covariance n'est donc pas inversible et n'importe quelle composante peut être supprimée sans perdre d'information qui pourrait être utilisée pour calculer la vraisemblance. Considérons $\sqrt{n_c}\underline{\boldsymbol{\varepsilon}}$, c'est le même vecteur que $\sqrt{n_c}\boldsymbol{\varepsilon}$ avec la dernière composante enlevée, et donc, sa matrice de covariance est la même avec la dernière ligne et la dernière colonne enlevées : $\boldsymbol{\Sigma} = diag(\underline{\boldsymbol{p}}) - \underline{\boldsymbol{p}}\underline{\boldsymbol{p}}^T$ ($\underline{\boldsymbol{p}}$ est $\boldsymbol{p}$ avec le dernier élément enlevé). Elle peut être estimée avec les probabilités d'échantillonnage $\underline{\widehat{\boldsymbol{p}}} = \begin{pmatrix} \widehat{p_1} \\ \vdots \\ \widehat{p_{d-1}} \end{pmatrix}$ au lieu de $\underline{\boldsymbol{p}}$. La matrice résultante est $\widehat{\boldsymbol{\Sigma}} = diag(\underline{\widehat{\boldsymbol{p}}}) - \underline{\widehat{\boldsymbol{p}}}\underline{\widehat{\boldsymbol{p}}}^T$. Il est facile de vérifier que si aucun élément de $\widehat{\boldsymbol{p}} = \begin{pmatrix} \widehat{p_1} \\ \vdots \\ \widehat{p_d} \end{pmatrix}$ (avec $\widehat{p}_d = 1 - \sum_{k=1}^{d-1} \widehat{p}_k$) n'est nul, alors $\widehat{\boldsymbol{\Sigma}}$ est inversible et

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \frac{1}{\widehat{p}_d}\mathbb{1} + diag(1/\underline{\widehat{\boldsymbol{p}}}) \tag{A.2}$$

est son inverse. $1/\underline{\widehat{\boldsymbol{p}}}$ est l'inverse de $\underline{\widehat{\boldsymbol{p}}}$ élément par élément et $\mathbb{1}$ est la matrice $d-1 \times d-1$ entièremenet remplies de 1. En pratique, les éléments de $\widehat{\boldsymbol{p}}$ peuvent être des zéros, ce qui rendrait la matrice singulière. Afin de contourner ce problème et d'éviter de donner trop d'importance aux erreurs sur les résultats rarement observés, nous modifions la probabilité d'échantillonnage et créons un nouveau vecteur $\widetilde{\boldsymbol{p}}$:

$$\widetilde{\boldsymbol{p}} = \frac{\widehat{\boldsymbol{p}} + \frac{5}{n_c}}{1 + \frac{5d}{n_c}}. \tag{A.3}$$

Cela signifie que nous considérons que chaque résultat a été observée 5 fois de plus qu'il ne l'a été en réalité, et que le nombre total d'observations passe de n_c à $n_c + 5d$ (le choix de 5 est arbitraire, il sera partiellement justifié en fin de section). Il s'agit d'une méthode standard pour rendre un critère plus lisse (voir [BK10]). L'estimation résultante de l'inverse de la matrice de covariance est la suivante :

$$\widetilde{\boldsymbol{\Sigma}}^{-1} = \frac{1}{\widetilde{p}_d}\mathbb{1} + diag(1/\underline{\widetilde{\boldsymbol{p}}}). \tag{A.4}$$

Avec l'inverse de $\widetilde{\boldsymbol{\Sigma}}$ et sachant que la distribution est normale et centrée, nous pouvons calculer la log-vraisemblance négative du vecteur (voir [Gut09]) :

$$\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = n_c \underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')^T \widetilde{\boldsymbol{\Sigma}}^{-1} \underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}'). \tag{A.5}$$

Nous utilisons $\widehat{\boldsymbol{p}}$ et non $\widetilde{\boldsymbol{p}}$ pour calculer $\underline{\boldsymbol{\varepsilon}}$. Si on utilisait $\widetilde{\boldsymbol{p}}$ l'estimateur qui minimise le critère serait biaisé (car le minimum de $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}$ "essayerait de faire coller" les probabilité à $\widetilde{\boldsymbol{p}}$ qui ne contient pas les probabilités empiriques observées) et le critère ne serait pas plus "lisse".

Simplifions cette expression en utilisant (A.4) et le fait que $\sum_k \varepsilon_k = 0 \Rightarrow \varepsilon_d = -\sum_{k=1}^{d-1} \varepsilon_k$:

$$\begin{aligned}
n_c \underline{\boldsymbol{\varepsilon}}^T \widetilde{\boldsymbol{\Sigma}}^{-1} \underline{\boldsymbol{\varepsilon}} &= n_c \underline{\boldsymbol{\varepsilon}}^T \begin{pmatrix} \frac{1}{\widetilde{p}_d} \sum_{k=1}^{d-1} \varepsilon_k + \frac{\varepsilon_1}{\widetilde{p}_1} \\ \vdots \\ \frac{1}{\widetilde{p}_d} \sum_{k=1}^{d-1} \varepsilon_k + \frac{\varepsilon_{d-1}}{\widetilde{p}_{d-1}} \end{pmatrix} \\
&= n_c \underline{\boldsymbol{\varepsilon}}^T \begin{pmatrix} \frac{\varepsilon_1}{\widetilde{p}_1} - \frac{\varepsilon_d}{\widetilde{p}_d} \\ \vdots \\ \frac{\varepsilon_{d-1}}{\widetilde{p}_{d-1}} - \frac{\varepsilon_d}{\widetilde{p}_d} \end{pmatrix} \\
&= n_c \big(\sum_{k=1}^{d-1} \frac{\varepsilon_k^2}{\widetilde{p}_k} - \frac{\varepsilon_d}{\widetilde{p}_d} \sum_{k=1}^{d-1} \varepsilon_k \big) \\
&= n_c \sum_{k=1}^{d} \frac{\varepsilon_k^2}{\widetilde{p}_k}.
\end{aligned}$$

L'expression de la log-vraisemblance négative est donc :

$$\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = n_c \sum_{k=1}^{d} \frac{\varepsilon_k(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')^2}{\widetilde{p}_k}. \tag{A.6}$$

On retrouve l'expression des statistiques du test du χ^2 de Pearson (avec $\widetilde{p}_k$ remplacé par p_k car il n'y a pas de régularisation dans le test de Pearson). Le test du χ^2 de Pearson vise à vérifier qu'une variable aléatoire continue suit une loi donnée. L'idée de calculer l'histogramme (avec un nombre de catégories donné qui correspond à nos d résultats possibles) des réalisations observées, et de comparer les fréquences d'occurrences théoriques (d'après la loi à tester) et empiriques de chaque catégorie de l'histogramme. Les carrés des écarts entre les fréquences théoriques et empiriques sont ensuite divisés par leurs écarts types, sommés (comme (A.6)), et [Pea00] montre que cette somme suit une loi de χ^2 à $d-1$ degrés de liberté.

La principale différence avec ce que nous faisons ici est que nous voulons trouver la probabilité théorique qui maximise la vraisemblance, alors que le test de Pearson suppose que la probabilité théorique est connue et fixe et vise à tester si la différence entre les deux probabilités peut être due au hasard et à donner une "p-value". Notre travail pour arriver à (A.6) n'est donc pas en vain. En effet, montrer que la somme des élément d'un vecteur suit une loi du χ^2 ne suffit pour montrer que le vecteur est gaussien, et calculer sa vraisemblance.

Le lien avec le test de Pearson justifie a posteriori le choix de la constante 5 dans (A.3). En effet, la fiabilité du test de Pearson dans le cas où un résultat a été trop peu observé a été étudiée dans la communauté des statisticiens. Et il est largement accepté que le nombre d'occurrences de chaque résultat doit être plus grand que 5 pour que l'approximation gaussienne ne soit pas aberrante (voir le deuxième paragraphe de l'introduction de [Yat34]).

A.2.3 Extension à plusieurs à la matrice de covariance et la vraisembance avec les résultats de plusieurs types mesures

Depuis le début de l'annexe, nous avons supposé qu'un seul type de mesure avec d résultats était effectué. En pratique, les méthodes que nous décrivons nécessitent 4 (dans la section 2.2) ou $2n_{qb}+1$ (dans la section 2.3) types de mesures. Les erreurs entre les probabilités empiriques et théoriques des différentes mesures sont indépendantes. Par conséquent, si $\underline{\boldsymbol{\varepsilon}}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')$ contient $n_t > 1$ types de mesures et dn_t composantes réelles, sa matrice de covariance est une matrice diagonale par bloc avec la matrice de covariance de chaque type de mesure sur la diagonale (parce que les erreurs de mesure sur deux types de mesure différents sont indépendantes). Il en

va de même pour l'inverse de sa matrice de covariance régularisée :

$$\widetilde{\mathbf{\Sigma}}^{-1} = \begin{bmatrix} \widetilde{\mathbf{\Sigma}}_1^{-1} & & \\ & \ddots & \\ & & \widetilde{\mathbf{\Sigma}}_{n_t}^{-1} \end{bmatrix}. \tag{A.7}$$

Chaque $\widetilde{\mathbf{\Sigma}}_k^{-1}$ est l'inverse régularisé de la matrice de covariance pour un type de mesure défini dans (A.4).

La log-vraisemblance négative des erreurs de $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')$ contenu $n_{prob} = n_t d$ mesures sur les n_t types de mesures est la somme des n_t log-vraisemblances négatives des vecteurs d'erreur de chaque type de mesure.

$$\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = n_c \sum_{k=1}^{n_{prob}} \frac{\varepsilon_k(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')^2}{\widetilde{p}_k}. \tag{A.8}$$

A.3 Algorithme PhaseCut

Entrée : Matrice $\mathbf{M}$ de la section 2.2.3

---

1 : $\mathbf{U}^0 = \mathbf{I}_d$, $N_{run} = 5000$
2 : for $j = 1, ..., N_{run}$
3 : $\quad \mathbf{U}^k = \mathbf{U}^{k-1}$
4 : $\quad$ choisir $k \in \{1, ..., d\}$ (aléatoire uniform). $\boldsymbol{k}_c = [1, ..., k-1, k+1, d]^T$
5 : $\quad$ calculer $\boldsymbol{u} = \mathbf{U}^j_{\boldsymbol{k}_c,\boldsymbol{k}_c} \boldsymbol{m}_{\boldsymbol{k}_c,k}$ et $\gamma = \boldsymbol{u}^* \boldsymbol{m}_{\boldsymbol{k}_c,k}$ avec $\boldsymbol{m}_{\boldsymbol{k}_c,k}$ la k-ème colonne de $\mathbf{M}$ où le k-ème élément a été retiré. $\mathbf{U}^j_{\boldsymbol{k}_c,\boldsymbol{k}_c}$ est $\mathbf{U}^j$ avec la k-ème ligne et la k-ème colonne enlevées.
6 : $\quad$ Si $\gamma > 0$ alors $\boldsymbol{u}^{k+1}_{\boldsymbol{k}_c,k} = \boldsymbol{u}^{k+1^*}_{k,\boldsymbol{k}_c} = -\sqrt{\frac{1}{\gamma}}\boldsymbol{u}$ sinon $\boldsymbol{u}^{k+1}_{\boldsymbol{k}_c,k} = \boldsymbol{u}^{k+1^*}_{k,\boldsymbol{k}_c} = 0$ avec $\boldsymbol{u}^{k+1}_{\boldsymbol{k}_c,k}$ et $\boldsymbol{u}^{k+1}_{k,\boldsymbol{k}_c}$ qui sont dans $\mathbf{U}^k$ (k-ème col. sans le k-ème elt. et k-ème lig. sans le k-ème elt. resp.)
7 : end for

Sortie : La matrice $\mathbf{U}^{N_{run}}$ hermitienne positive et qui contient seulement des 1 sur la diagonale

---

A.4 Lien entre la fidélité et notre métrique d'erreur

La fidélité est définie par (1.9) : $f(\boldsymbol{\rho}, \widehat{\boldsymbol{\rho}}) = tr\left(\sqrt{\sqrt{\boldsymbol{\rho}}\widehat{\boldsymbol{\rho}}\sqrt{\boldsymbol{\rho}}}\right)$. Si les deux états sont purs, alors $\boldsymbol{\rho} = \boldsymbol{v}\boldsymbol{v}^*$ et $\widehat{\boldsymbol{\rho}} = \widehat{\boldsymbol{v}}\widehat{\boldsymbol{v}}^*$ sont de rang 1, et de trace 1. Elle ont donc une seule valeur propre non nulle, et celle-ci vaut 1. Donc $\sqrt{\boldsymbol{\rho}} = \boldsymbol{\rho}$, où la racine carrée d'une matrice hermitienne positive est définie avec la décomposition spectrale (même vecteurs propres, racine carrée des valeurs propres), il en va de même pour $\widehat{\boldsymbol{\rho}}$. Donc :

$f(\boldsymbol{\rho}, \widehat{\boldsymbol{\rho}}) = tr\left(\sqrt{\boldsymbol{\rho}\widehat{\boldsymbol{\rho}}\boldsymbol{\rho}}\right) = tr\left(\sqrt{\boldsymbol{v}(\boldsymbol{v}^*\widehat{\boldsymbol{v}})(\widehat{\boldsymbol{v}}^*\boldsymbol{v})\boldsymbol{v}^*}\right) = tr(\sqrt{|\boldsymbol{v}^*\widehat{\boldsymbol{v}}|^2\boldsymbol{\rho}}) = |\boldsymbol{v}^*\widehat{\boldsymbol{v}}| tr(\boldsymbol{\rho}) = |\boldsymbol{v}^*\widehat{\boldsymbol{v}}|$.

Considérons maintenant notre erreur : $\mu_s(\boldsymbol{v}, \widehat{\boldsymbol{v}}) = \frac{1}{\sqrt{2}}||\boldsymbol{v} - \widehat{\boldsymbol{v}}.e^{-i\xi}||_2$ où $\xi \in [0, 2\pi]$ est l'angle qui minimise μ_s (ou μ_s^2). Développons μ_s^2 : $\mu_s^2 = \frac{1}{2}||\boldsymbol{v} - \widehat{\boldsymbol{v}}.e^{-i\xi}||_2^2 = \frac{1}{2}\left(||\boldsymbol{v}||_2^2 + ||\widehat{\boldsymbol{v}}.e^{-i\xi}||_2^2 - 2Re(e^{-i\xi}\boldsymbol{v}^*\widehat{\boldsymbol{v}})\right) =$

$1 - Re(e^{-i\xi}\boldsymbol{v}^*\widehat{\boldsymbol{v}})$ Sous cette forme, il est clair que la valeur de ξ qui minimise la métrique est $Arg\left(\frac{\overline{\boldsymbol{v}^*\widehat{\boldsymbol{v}}}}{|\boldsymbol{v}^*\widehat{\boldsymbol{v}}|}\right)$, et $Re(e^{-i\xi}\boldsymbol{v}^*\widehat{\boldsymbol{v}}) = |\boldsymbol{v}^*\widehat{\boldsymbol{v}}| = f$. Donc $\mu_s^2 = 1 - f \Rightarrow f = (1 - \mu_s^2)$

Annexe B

Annexe du chapitre 3

Dans cette annexe, nous donnons les démonstrations de résultats du Chapitre 3.

B.1 Solution du problème de moindres carrés totaux sous contrainte d'unitarité

Trouver le $\widehat{\mathbf{U}}$ qui satisfait (3.11) est équivalent à :

$$\widehat{\mathbf{U}}_{LS} = \underset{\widehat{\mathbf{U}} \in \mathbb{U}_d(\mathbb{C})}{\operatorname{argmin}} \left(\min_{\widehat{\mathbf{X}} \in \mathbb{C}^{d \times n}} \left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}} \end{bmatrix} \right\|_2^2 \right). \tag{B.1}$$

avec $\Delta_{\mathbf{X}} = \widehat{\mathbf{X}} - \widetilde{\mathbf{X}}$ et $\Delta_{\mathbf{X}} = \widetilde{\mathbf{Y}} - \widehat{\mathbf{U}}\widehat{\mathbf{X}}$

Nous choisissons de diviser le problème en 2 :

1. Pour toute matrice unitaire $\widehat{\mathbf{U}}$, on cherche le $\mathbf{X}_{\widehat{\mathbf{U}}}$ (et $\mathbf{Y}_{\widehat{\mathbf{U}}} = \widehat{\mathbf{U}}\mathbf{X}_{\widehat{\mathbf{U}}}$) solutions de :

$$\mathbf{X}_{\widehat{\mathbf{U}}} = \underset{\widehat{\mathbf{X}}}{\operatorname{argmin}} \left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}} \end{bmatrix} \right\|_2^2. \tag{B.2}$$

2. Toute $\widehat{\mathbf{U}}_{LS}$ solution de (B.1) est aussi solution de :

$$\widehat{\mathbf{U}}_{LS} = \underset{\widehat{\mathbf{U}} \in \mathbb{U}_d(\mathbb{C})}{\operatorname{argmin}} \left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} \widehat{\mathbf{X}}_{\widehat{\mathbf{U}}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}}_{\widehat{\mathbf{U}}} \end{bmatrix} \right\|_2^2. \tag{B.3}$$

Résolvons (B.2)

$$\left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}} \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} \right\|_2^2 + \left\| \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}} \end{bmatrix} \right\|_2^2 - 2Re\left(tr\left(\begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix}^H \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}} \end{bmatrix} \right) \right)$$

$$= \left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} \right\|_2^2 + \left\| \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{U}}\widehat{\mathbf{X}} \end{bmatrix} \right\|_2^2 - 2Re\big(tr(\widetilde{\mathbf{X}}^H\widehat{\mathbf{X}} + \widetilde{\mathbf{Y}}^H\widehat{\mathbf{U}}\widehat{\mathbf{X}})\big) = C_1 + 2\|\widehat{\mathbf{X}}\|_2^2 - 2Re\Big(tr\big((\widetilde{\mathbf{X}} + \widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}})^H\widehat{\mathbf{X}}\big)\Big)$$

$$= C_1 + 2\Big(\|\widehat{\mathbf{X}}\|_2^2 - 2Re\Big(tr\big((\tfrac{1}{2}\widetilde{\mathbf{X}} + \tfrac{1}{2}\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}})^H\widehat{\mathbf{X}}\big)\Big)\Big)$$

avec $C_1 = \|\widetilde{\mathbf{X}}\|_2^2 + \|\widetilde{\mathbf{Y}}\|_2^2$

La quantité à minimiser dans (B.2) s'écrit donc $C_2 + 2\left\|\widehat{\mathbf{X}} - (\frac{1}{2}\widetilde{\mathbf{X}} + \frac{1}{2}\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}})\right\|_2^2$

avec $C_2 = C_1 - 2\|\frac{1}{2}\widetilde{\mathbf{X}} + \frac{1}{2}\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}}\|_2^2$

C_1 et C_2 ne dépendent pas $\widehat{\mathbf{X}}$.
avec ces expressions, la solution (unique) de (B.2) est évidente est : $\widehat{\mathbf{X}}_{\widehat{\mathbf{U}}} = \frac{1}{2}\widetilde{\mathbf{X}} + \frac{1}{2}\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}}$ et donc$\widehat{\mathbf{Y}}_{\widehat{\mathbf{U}}} = \frac{1}{2}\widetilde{\mathbf{Y}} + \frac{1}{2}\widehat{\mathbf{U}}\widetilde{\mathbf{X}}$

Résolvons maintenant (B.3) :

$$\left\| \begin{bmatrix} \widetilde{\mathbf{X}} \\ \widetilde{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{Y}} \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} \frac{1}{2}\widetilde{\mathbf{X}} - \frac{1}{2}\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}} \\ \frac{1}{2}\widetilde{\mathbf{Y}} - \frac{1}{2}\widehat{\mathbf{U}}\widetilde{\mathbf{X}} \end{bmatrix} \right\|_2^2 = \frac{1}{4}\Big(2\|\widetilde{\mathbf{X}}\|_2^2 + 2\|\widetilde{\mathbf{Y}}\|_2^2 - 2Re(tr(\widetilde{\mathbf{X}}^H\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}} + \widetilde{\mathbf{Y}}^H\widehat{\mathbf{U}}\widetilde{\mathbf{X}}))\Big)$$
$$= \frac{1}{4}\Big(C_3 - 4Re(tr(\widetilde{\mathbf{X}}^H\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}}))\Big) = \frac{1}{4}\Big(C_3 - 4Re(tr(\widehat{\mathbf{U}}^H\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H))\Big)$$

où $C_3 = 2\|\widetilde{\mathbf{X}}\|_2^2 + 2\|\widetilde{\mathbf{Y}}\|_2^2$ ne dépend pas de $\widehat{\mathbf{U}}$. Ldéfinissons la décomposition en valeur singulière (voir section B.2 pour son unicité) de $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^H$. L'optimum $\widehat{\mathbf{U}}$ minimise la quantité suivante : $-Re(tr(\widehat{\mathbf{U}}^H\mathbf{P}\mathbf{\Sigma}\mathbf{Q}^H)) = -Re(tr(\mathbf{Q}^H\widehat{\mathbf{U}}^H\mathbf{P}\mathbf{\Sigma}))$.

$\mathbf{Q}^H\widehat{\mathbf{U}}^H\mathbf{P}$ est une matrice unitaire, définissons ses colonnes : $\mathbf{Q}^H\widehat{\mathbf{U}}^H\mathbf{P} = \begin{bmatrix}\mathbf{o}_1 & ... & \mathbf{o}_k\end{bmatrix}$, et soient $\sigma_1, ..., \sigma_d$ et $o^r_{1,1}, ..., o^r_{d,d}$ les composantes de la diagonales de $\mathbf{\Sigma}$ et de $Re(\mathbf{Q}^H\widehat{\mathbf{U}}^H\mathbf{P})$ respectivement. Tous les $\{o^r_{j,j}\}_j$ sont inférieurs à 1 (ce sont les parties réelles d'éléments d'une matrice unitaire). Avec ces notations :

$$\begin{array}{rcl} -Re(tr(\mathbf{Q}^H\widehat{\mathbf{U}}^H\mathbf{P}\mathbf{\Sigma})) & = & -Re(tr(\begin{bmatrix}\sigma_1\mathbf{o}_1 & ... & \sigma_d\mathbf{o}_d\end{bmatrix})) \\ & = & -\sum_{j=1}^d \delta_j o^r_{j,j} \\ & \geq & -\sum_{j=1}^d \delta_j. \end{array},$$

et le minimum est seulement atteint pour $\mathbf{Q}^H\widehat{\mathbf{U}}^H\mathbf{P} = \mathbf{I}$ (car c'est la seule matrice unitaire qui n'a que des 1 sur sa diagonale). Donc $\widehat{\mathbf{U}}_{LS} = \mathbf{P}\mathbf{Q}^H$ minimise la norme au carré de l'erreur entre les mesures et les estimées de $\mathbf{X}$ et $\mathbf{Y}$. Ce minimum est unique sauf si il existe plusieurs décompositions en valeurs singulières de $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$ qui donnent des $\mathbf{P}\mathbf{Q}^H$ différents. Les estimés associés des $\mathbf{X}$ et $\mathbf{Y}$ sont :

$$\widehat{\mathbf{X}}_{\widehat{\mathbf{U}}_{LS}} = \frac{1}{2}\widetilde{\mathbf{X}} + \frac{1}{2}\widehat{\mathbf{U}}_{LS}^H\widetilde{\mathbf{Y}} \quad \widehat{\mathbf{Y}}_{\widehat{\mathbf{U}}_{LS}} = \frac{1}{2}\widetilde{\mathbf{Y}} + \frac{1}{2}\widehat{\mathbf{U}}_{LS}\widetilde{\mathbf{X}} \tag{B.4}$$

B.2 Preuve de l'unicité

Dans la présente section, nous montrons que le minimum est unique si et seulement si $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$ est inversible ($\widetilde{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ sont définis dans B.1). Ce qui est équivalent à "$\widetilde{\mathbf{X}}$ et $\widetilde{\mathbf{Y}}$ sont de rang plein et $k \geq d$".

Les minima que nous avons trouvés dans la section précédente pour (B.2) et (B.3) sont des minima globaux stricts sauf si la matrice unitaire $\mathbf{P}\mathbf{Q}^H$ où $\mathbf{P}$ et $\mathbf{Q}$ sont issus de la décomposition en valuers singulière de $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$ n'est pas unique. Les valeurs singulières d'une matrice sont toujours uniques, mais $\mathbf{P}$ et $\mathbf{Q}$ ne le sont pas. Nous considérons que les valeurs singulières sont triées dans l'ordre croissant sur la diagonale de $\mathbf{\Sigma}$. Cela n'a pas d'importance car, si nous changeons l'ordre et permutons les colonnes associées de $\mathbf{P}$ et $\mathbf{Q}$, alors $\mathbf{P}\mathbf{Q}^H$ ne change pas. Par conséquent, $\mathbf{\Sigma}$ peut être considérée comme unique (valeurs singulières uniques dans un ordre croissant sur la diagonale). Définissons l'ensemble $\mathcal{E}$ qui contient tous les $\mathbf{P}$ et $\mathbf{Q}$ unitaires admissibles pour la décomposition en valeurs singulières de $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$:

$$\mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}} = \Big\{\{\mathbf{P} \in \mathbb{U}_d, \mathbf{Q} \in \mathbb{U}_d\}, \text{ t.q. } \widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^H\Big\}.$$

Avec cette définition, "$\mathbf{P}\mathbf{Q}^H$ est unique" peut être traduit formellement en "$\Big\{\mathbf{P}\mathbf{Q}^H, \text{ t.q. } \{\mathbf{P},\mathbf{Q}\} \in \mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\Big\}$ est un singleton".

Lemme $\Big\{\mathbf{P}\mathbf{Q}^H, \text{ t.q. } \{\mathbf{P},\mathbf{Q}\} \in \mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\Big\}$ est un singleton si et seulement si $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$ est inversible.

Démonstration. L'implication réciproque ("seulement si") est plus simple à montrer, nous allons la montrer par contraposition. Si $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$ n'est pas inversible, alors sa première (et plus petite)

valeur singulière est 0, donc, les premiers éléments de la diagonale de $\mathbf{\Sigma}$ sont 0. Soit $\{\mathbf{P}, \mathbf{Q}\}$ un élément de $\mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}$, $\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^H$. Soit $\{\boldsymbol{p}_j\}_j$ une colonne de $\mathbf{P}$, définissons :

$$\mathbf{P}_2 = \begin{bmatrix} \boldsymbol{p}_1 e^{i\theta} & \boldsymbol{p}_2 & \dots & \boldsymbol{p}_k \end{bmatrix} \tag{B.5}$$

avec θ un angle de $]0, 2\pi[$. Comme le premier élément de la diagonale de $\mathbf{\Sigma}$ est 0, on peut facilement montrer que $\mathbf{P}_2\mathbf{\Sigma}\mathbf{Q}^H = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^H$, et donc $\{\mathbf{P}_2, \mathbf{Q}\}$ est aussi dans $\mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}$. Or, comme θ est un angle non nul, $\mathbf{P}_2\mathbf{Q}^H \neq \mathbf{P}\mathbf{Q}^H$. Donc, $\mathbf{P}_2\mathbf{Q}^H$ et $\mathbf{P}_2\mathbf{Q}^H$ sont deux éléments distincts de $\left\{\mathbf{P}\mathbf{Q}^H, \text{ t.q. } \{\mathbf{P}, \mathbf{Q}\} \in \mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}\right\}$, l'ensemble n'est donc pas un singleton.

Montrons maintenant l'implication directe ("si"). Soient $\{\mathbf{P}_1, \mathbf{Q}_1\}$ et $\{\mathbf{P}_2, \mathbf{Q}_2\}$ deux paires de matrices de $\mathcal{E}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}}$. Par définition, $\mathbf{P}_1\mathbf{\Sigma}\mathbf{Q}_1^H = \mathbf{P}_2\mathbf{\Sigma}\mathbf{Q}_2^H$, et nous voulons montrer que, si la matrice diagonale $\mathbf{\Sigma}$ ne contient que des éléments strictement positifs sur la diagonale (équivalent à "$\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H$ est inversible"), alors $\mathbf{P}_1\mathbf{Q}_1^H = \mathbf{P}_2\mathbf{Q}_2^H$. Pour ce faire, nous allons montrer que pour tout couple de colonne $\boldsymbol{p}_{(k,1)}\boldsymbol{q}_{(j,1)}$ de $\mathbf{P}_1$ et $\mathbf{Q}_1$ respectivement, on a :

$$\boldsymbol{p}_{(k,1)}^H\mathbf{P}_1\mathbf{Q}_1^H\boldsymbol{q}_{(j,1)} = \boldsymbol{p}_{(k,1)}^H\mathbf{P}_2\mathbf{Q}_2^H\boldsymbol{q}_{(j,1)}. \tag{B.6}$$

C'est équivalent à $\mathbf{P}_1^H\mathbf{P}_1\mathbf{Q}_1^H\mathbf{Q}_1 = \mathbf{P}_1^H\mathbf{P}_2\mathbf{Q}_2^H\mathbf{Q}_1$ qui est équivalent à $\mathbf{P}_1\mathbf{Q}_1^H = \mathbf{P}_2\mathbf{Q}_2^H$ car $\mathbf{P}_1^H$ et $\mathbf{Q}_1$ sont inversibles.

Nous utiliserons le fait que les colonnes de $\mathbf{P}_1$ et celles de $\mathbf{P}_2$ (resp. les colonnes de $\mathbf{Q}_1$ et celles de $\mathbf{Q}_2$) sont des vecteurs propres de $\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)^H$ (resp. de $\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)^H\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)$) et la matrice diagonale qui contient les valeurs propres est $\mathbf{\Sigma}^2$, ou formellement :

$$\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)^H = \mathbf{P}_1\mathbf{\Sigma}^2\mathbf{P}_1^H = \mathbf{P}_2\mathbf{\Sigma}^2\mathbf{P}_2^H$$
$$\text{et } \left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right)^H\left(\widetilde{\mathbf{Y}}\widetilde{\mathbf{X}}^H\right) = \mathbf{Q}_1\mathbf{\Sigma}^2\mathbf{Q}_1^H = \mathbf{Q}_2\mathbf{\Sigma}^2\mathbf{Q}_2^H.$$

Et même si les vecteurs propres d'une matrice donnée ne sont pas uniques, les espaces propres le sont, et ils sont orthogonaux entre eux pour les matrices hermitiennes. Cela signifie que, pour une colonne donnée $\boldsymbol{p}_{(k,1)}$ de $\mathbf{P}_1$ et une colonne donnée $\boldsymbol{q}_{(j,1)}$ de $\mathbf{Q}_1$ associées aux valeurs propres σ_k^2 et σ_j^2 respectivement (ou, de manière équivalente, aux valeurs singulières σ_k et σ_j respectivement), on a

$$\mathbf{P}_2^H\boldsymbol{p}_{(k,1)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{P}_{2,k}^H\boldsymbol{p}_{(k,1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ et } \mathbf{Q}_2^H\boldsymbol{q}_{(j,1)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{Q}_{2,j}^H\boldsymbol{q}_{(j,1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

où $\mathbf{P}_{2,k}$ (resp. $\mathbf{Q}_{2,j}$) contient les colonnes de $\mathbf{P}_2$ (resp. $\mathbf{Q}_2$) qui sont associés à la valeur singulière σ_k (resp. σ_j) dans l'ordre dans lequel ils aparraissent dans $\mathbf{P}_2$ (resp. $\mathbf{Q}_2$).

Soit $\boldsymbol{p}_{(k,1)}$ et $\boldsymbol{q}_{(j,1)}$ des colonnes de $\mathbf{P}_1$ et $\mathbf{Q}_1$ respectivement. Montrons (B.6) en utilisant le fait que $\mathbf{P}_1\mathbf{\Sigma}\mathbf{Q}_1^H = \mathbf{P}_2\mathbf{\Sigma}\mathbf{Q}_2^H$.

$$
\begin{aligned}
\mathbf{P}_1\mathbf{\Sigma}\mathbf{Q}_1^H = \mathbf{P}_2\mathbf{\Sigma}\mathbf{Q}_2^H \quad &\Rightarrow \quad \boldsymbol{p}_{(k,1)}^H\mathbf{P}_1\mathbf{\Sigma}\mathbf{Q}_1^H\boldsymbol{q}_{(j,1)} = \boldsymbol{p}_{(k,1)}^H\mathbf{P}_2\mathbf{\Sigma}\mathbf{Q}_2^H\boldsymbol{q}_{(j,1)} \\
&\Rightarrow \quad \boldsymbol{\delta}_k^H\mathbf{\Sigma}\boldsymbol{\delta}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{P}_{2,k}^H\boldsymbol{p}_{(k,1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^H \mathbf{\Sigma} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{Q}_{2,j}^H\boldsymbol{q}_{(j,1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&\Rightarrow \quad \sigma_j\delta_{j,k} = \sigma_j\delta_{j,k}\boldsymbol{p}_{(k,1)}^H\mathbf{P}_{2,k}\mathbf{Q}_{2,j}^H\boldsymbol{q}_{(j,1)} \\
&\Rightarrow \quad \delta_{j,k} = \delta_{j,k}\boldsymbol{p}_{(k,1)}^H\mathbf{P}_{2,k}\mathbf{Q}_{2,j}^H\boldsymbol{q}_{(j,1)} \text{ car } (\sigma_j \neq 0)
\end{aligned}
$$

où $\boldsymbol{\delta}_k$ est le vecteur colonne à d dimension $(d-1)$ zeros et un 1 sur le k-ème élément, et $\delta_{j,k}$ est 1 si $j = k$ et 0 sinon. Le même calcul montre que $\boldsymbol{p}_{(k,1)}^H\mathbf{P}_1\mathbf{Q}_1^H\boldsymbol{q}_{(j,1)} = \delta_{j,k}$ et $\boldsymbol{p}_{(k,1)}^H\mathbf{P}_2\mathbf{Q}_2^H\boldsymbol{q}_{(j,1)} = \delta_{j,k}\boldsymbol{p}_{(k,1)}^H\mathbf{P}_{2,k}\mathbf{Q}_{2,j}^H\boldsymbol{q}_{(j,1)}$. Nous avons donc montré (B.6) : $\boldsymbol{p}_{(k,1)}^H\mathbf{P}_1\mathbf{Q}_1^H\boldsymbol{q}_{(j,1)} = \boldsymbol{p}_{(k,1)}^H\mathbf{P}_2\mathbf{Q}_2^H\boldsymbol{q}_{(j,1)}$ cqfd.

■

B.3 Preuve de la condition d'identifiabilité

Notre condition d'identifiabilité est (3.19). Elle est nécessaire et suffisante pour que $\mathbf{M}$ soit identifiable (à une phase globale près et avec une configuration de QPT donnée) dans l'ensemble des matrices unitaires. Dans B.3.1, nous montrons qu'il s'agit d'une condition suffisante en montrant que notre algorithme est capable de retrouver $\mathbf{M}$ à une phase globale près. Cela qui signifie que notre algorithme de QPT fonctionne dans toutes les situations où la QPT est théoriquement possible. Dans B.3.2, nous montrons que (3.19) est condition nécessaire.

Dans [RGK13], Reich et al. ont leur propre condition nécessaire et suffisante sur les états d'entrée, ils montrent qu'elle garantit que le processus représenté par $\mathbf{M}$ est identifiable, pas simplement parmi les processus unitaires, mais parmi tous les processus (unitaires ou non). Dans B.3.3, nous montrons que leur condition est équivalente à (3.3).

La preuve de B.3.3 rend B.3.1 quelque peu redondants, parce que la définition de l'identifiabilité dans Reich et al. est plus forte que la nôtre. Nous incluons quand même B.3.1 parce qu'elle montre le caractère suffisant de (3.19) expliquant pourquoi notre algorithme fonctionne toujours si (3.19) est satisfaite, ce qui est un résultat important en soi.

B.3.1 Preuve que (3.19) est suffiante

Afin de montrer que (3.19) est suffisant pour que QPT soit possible (ou "$\mathbf{M}$ soit identifiable") avec la configuration de la figure 3.1, nous devons définir ce que nous entendons par "QPT est possible" (ou "$\mathbf{M}$ est identifiable"). Formellement, cela signifie que les sorites de la QST ($\widehat{\mathbf{X}}$ et $\widehat{\mathbf{Y}}$) ne sont compatibles que avec une seule matrice $\mathbf{M}$ (à une phase près) qui rerésente le processus. On rappelle le lien de (3.18) entre la matrice $\mathbf{M}$ et les sorties de la QST quand il n'y a pas d'erreur de QST : $\widehat{\mathbf{Y}}\mathbf{D}(\boldsymbol{\xi}) = \mathbf{M}\widehat{\mathbf{X}}$, avec $\mathbf{D}(\boldsymbol{\xi}) = \begin{pmatrix} e^{i\xi_1} & & \\ & \ddots & \\ & & e^{i\xi_{n_x}} \end{pmatrix}$. L'ensemble des matrices représentant des processus unitaires qui sont compatibles avec les résultats de la QST est :

$$
\mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}) = \left\{ \mathbf{U} \in \mathbb{U}_d, \text{s.t.} \exists \boldsymbol{\xi} \in \mathbb{R}^{n_x} \text{t.q.} \mathbf{U}\ \widehat{\mathbf{X}}\ \mathbf{D}(\boldsymbol{\xi})^* = \widehat{\mathbf{Y}} \right\}. \tag{B.7}
$$

Or il existe $\boldsymbol{\xi}_x$ et $\boldsymbol{\xi}_y$ tels que $\mathbf{X} = \widehat{\mathbf{X}}\mathbf{D}(\boldsymbol{\xi}_x)$ et $\mathbf{MX} = \widehat{\mathbf{Y}}\mathbf{D}(\boldsymbol{\xi}_x)$. Donc $\mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}) = \mathcal{U}(\mathbf{X}, \mathbf{MX})$ (il suffit de changer $\mathbf{D}(\boldsymbol{\xi})^*$ en $\mathbf{D}(\boldsymbol{\xi})^*\mathbf{D}(\boldsymbol{\xi}_x)\mathbf{D}(\boldsymbol{\xi}_y)^*$).

La QPT est donc possible si et seulement si $\mathcal{U}(\mathbf{X}, \mathbf{MX}) = \{e^{i\phi}\mathbf{M}\}_\phi$.

La condition de (3.19) est suffisante pour que la QPT soit possible, car si (3.19) est vraie, alors, nous allons montrer que les algorithmes de la section 3.1.2 et 3.1.3 donnent un résultat unique ($\mathbf{M}$) à une phase globale près :

- l'algorithme de la section 3.1.3 réussit toujours, c'est-à-dire que nous quittons l'algorithme à l'étape 4 ou 6. Cela découle de de l'égalité que nous avons soulignée entre le rang $\mathbf{F}_S^k(\widehat{\boldsymbol{x}}_{\ell_0})$ et la dimension du sous-espace généré par les colonnes de $\mathcal{S}$ après avoir passé l'étape k fois avec $b_{orth} = 0$. En effet $\text{rang}\,(\mathbf{F}_S^{n_x}(\boldsymbol{x}_{\ell_0})) = \text{rang}\,(\mathbf{F}_S^{n_x}(\widehat{\boldsymbol{x}}_{\ell_0}))$ (voir remarque après la définition de $\mathbf{F}_S$ dans (3.19)), donc, si $\text{rang}\,(\mathbf{F}_S^{n_x}(\boldsymbol{x}_{\ell_0})) = d$, alors, même si l'algorithme commence mal et que nous devons fixer $b_{orth} = 0$, la condition de l'étape 6 sera finalement satisfaite après avoir parcouru l'étape 5 au plus n_x fois. Le lecteur pourrait penser que nous pourrions quitter l'algorithme prématurément (c'est-à-dire avant que les éléments de $\mathcal{S}$ ne génèrent $\mathbb{C}^d$) parce que la condition qui nous fait boucler de l'étape 5 à l'étape 3 cesse d'être satisfaite. Ce n'est pas un problème, car si la condition n'est pas remplie, il est inutile de continuer, le nombre d'éléments de $\mathcal{S}$ n'augmentera pas même si nous devions aller à l'étape 3.
- Les $\widetilde{\mathbf{Y}}$ et $\widehat{\mathbf{X}}$ avec lesquels nous sortons de l'algorithme de la section 3.1.3 sont l'unique solution de (3.13) et les phases trouvées pour $\widetilde{\mathbf{Y}}$ sont les seules (avec $\xi_{l_0} = 0$) qui sont compatibles avec le modèle unitaire.
- Les matrices $\widetilde{\mathbf{Y}}$ et $\widehat{\mathbf{X}}$ avec lesquelles nous quittons l'algorithme de 3.1.3 sont de rangs pleins (d). Par conséquent, le problème des moindres carrés totaux avec contrainte unitaire a une solution unique $\mathbf{M}_{LS}$à une phase globale près (comme discuté à la fin de la section 3.1.2). Et comme les phases de $\widetilde{\mathbf{Y}}$ ont bien été corigées, $\mathbf{M}_{LS} = \mathbf{M}$.

Donc $\mathcal{U}(\mathbf{X}, \mathbf{MX}) = \{e^{i\phi}\mathbf{M}\}_\phi$.

B.3.2 Preuve que (3.19) est nécessaire

Montrons maintenant que (3.19) est une condition nécessaire. Nous supposons que (3.19) est fausse, c'est-à-dire que

$$\exists \ell \in \{1, ..., n_x\},\ \text{rang}\,(\mathbf{F}_S^{n_x}(\boldsymbol{x}_\ell)) < d \tag{B.8}$$

Listons quelques propriétes de la fonction suivante :
$\mathbf{G}_S : k \longrightarrow \mathbf{F}_S^k(\boldsymbol{x}_\ell)$:

- Les colonnes de $\mathbf{G}_S(k)$ sont aussi des colonnes de $\mathbf{G}_S(k+1)$.
- Si k_0 est le plus petit entier tel que $\mathbf{G}_S(k_0) = \mathbf{G}_S(k_0+1)$, alors $\forall k \geq k_0$, $\mathbf{G}_S(k_0) = \mathbf{G}_S(k)$.
- n_x est une borne supérieure pour le nombre de colonnes de $\mathbf{G}_S(k), \forall k$.

Par conséquent, le nombre de colonnes de $\mathbf{G}_S(k)$ augmente strictement avec k pour les premières k_0 itérations, et pour $k \geq k_0$ $\mathbf{G}_S$ devient constant. De plus, k_0 doit être plus petit que n_x car le nombre de colonnes augmente d'au moins 1 à chaque itération avant k_0 et est limité par n_x.

Pour tout ℓ, et en particulier pour le ℓ de (B.8), $\mathbf{F}_S^{n_x}(\boldsymbol{x}_\ell) = \mathbf{F}_S^{n_x+1}(\boldsymbol{x}_\ell)$. Par conséquent, nous pouvons diviser les colonnes de $\mathbf{X}$ en deux groupes : $\mathbf{X}_s$ (défini comme $\mathbf{F}_S^{n_x}(\boldsymbol{x}_\ell)$) et $\mathbf{X}_f$ (défini comme la matrice qui contient les autres colonnes dans l'ordre). La matrice $\mathbf{X}_f$ peut être vide, mais si elle ne l'est pas, les colonnes qu'elle contient sont toutes orthogonales aux colonnes de $\mathbf{X}_s$ (puisque $\mathbf{F}_S(\mathbf{X}_s) = \mathbf{X}_s$). D'après (B.8), $\mathbf{X}_s$ n'est pas de rang complet.

A partir de là, nous considérons les deux seules implications possibles de "(3.19) est fausse" :

1. $\mathbf{X}_f$ est vide, $\mathbf{X}_s = \mathbf{X}$ et $\text{rang}(\mathbf{X}) < d$.
2. $\mathbf{X}_f$ n'est pas vide $\mathbf{X}$ peut être décomposé : $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_f]\mathbf{Q}_{per}$, où $\mathbf{Q}_{per}$ est une matrice de permutation de taille $n_x \times n_x$.

Et chacune de ces conditions rend la QPT impossible car :

- Si la condition du Cas 1 est vraie, alors il existe un vecteur de norme unitaire appelé $\boldsymbol{v}_{ker}$ dans le kernel de $\mathbf{X}^*$, i.e. $\mathbf{X}^*\boldsymbol{v}_{ker} = \mathbf{0}$ et $\boldsymbol{v}_{ker}^*\mathbf{X} = \mathbf{0}$. Nous appellons $\mathbf{V}_{hker}$ une matrice de taille $d \times (d-1)$ telle que $\mathbf{P}_v = [\mathbf{V}_{hker}, \boldsymbol{v}_{ker}]$ soit une matrice unitaire ($\mathbf{V}_{hker}$ et $\mathbf{P}_v$ ne sont pas unique pour un $\boldsymbol{v}_{ker}$ donné, $\mathbf{V}_{hker}$ peut être n'importe quelle base orthogonale de l'hyperplan orthogonal à $\boldsymbol{v}_{ker}$). Nous définissons $\begin{bmatrix}\mathbf{C}_1 & \boldsymbol{c}_2\end{bmatrix} = \mathbf{M}\mathbf{P}_v$ ($\mathbf{C}_1 \in \mathbb{C}^{d\times(d-1)}, \boldsymbol{c}_2 \in \mathbb{C}^d$), il est facile de vérifier que pour tout angle ϕ t.q. $0 < \phi < 2\pi$, la matrice unitaire $\mathbf{M}_2 = \begin{bmatrix}\mathbf{C}_1 & \boldsymbol{c}_2 e^{i\phi}\end{bmatrix}\mathbf{P}_v^*$ n'est pas identique à $\mathbf{M}$ (même à une phase près car $d > 1$) et $\mathbf{M}$ et $\mathbf{M}_2$ sont toutes les deux dans $\mathcal{U}(\mathbf{X}, \widehat{\mathbf{Y}})$ car $\mathbf{M}\mathbf{X} = \mathbf{M}_2\mathbf{X}$.
- Si la condition du Cas 2 est vraie, nous n'avons qu'à considérer le cas où $\mathbf{X}$ est de rang plein (sinon, nous utilisons le raisonnement ci-dessus), et $vect(\mathbf{X}_f)$ est orthogonal au complément de $vect(\mathbf{X}_s)$ ($vect$ l'espace vectoriel généré par les colonnes d'une matrice). Nous définissons $\mathbf{P}_s$ (resp. $\mathbf{P}_f$) comme une matrice dont les colonnes une base orthogonale de $vect(\mathbf{X}_s)$ (resp. de $vect(\mathbf{X}_f)$). Nous définissons la matrice $\mathbf{M}_2$ comme $\mathbf{M}_2 = \mathbf{M}(\mathbf{P}_s\mathbf{P}_s^* e^{i\phi} + \mathbf{P}_f\mathbf{P}_f^*)$, et définissons $\mathbf{X}_{alt} = \begin{bmatrix}\mathbf{X}_s e^{-i\phi} & \mathbf{X}_f\end{bmatrix}\mathbf{Q}_{per}$ ($\mathbf{Q}_{per}$ défini ci-dessus). avec ces définitions $\mathbf{M}_2$ est unitaire (facile à vérifier), et un calcul simple montre que $\mathbf{M}\mathbf{X} = \mathbf{M}_2\mathbf{X}_{alt}$, $\mathbf{X}_{alt}$ a les mêmes colonnes que $\mathbf{X}$ à des phases près (pas les mêmes phrases sur chaque colonne), donc $\mathbf{M}$ et $\mathbf{M}_2$ sont toutes les deux dans $\mathcal{U}(\mathbf{X}, \mathbf{M}\mathbf{X})$ mais $\mathbf{M}_2$ et $\mathbf{M}$ ne sont pas identiques à une phase près.

B.3.3 Équivalence entre (3.19) et (3.26)

La condition de Reich et al. a été réécrite en (3.26), elle garantit l'identifiabilité du processus représenté par $\mathbf{M}$ parmi tous les processus. Cette garantie est plus forte que l'identifiabilité parmi les processus unitaires, que nous garantissons avec (3.19). Par conséquent, (3.26) implique (3.19). Montrons que (3.19) implique (3.26) :

Soit $\mathbf{C} \in \mathbb{U}_d(\mathbb{C})$ une matrice unitaire de $Com\left(\{\boldsymbol{x}_\ell \boldsymbol{x}_\ell^*\}_\ell\right)$. Montrons que si (3.19) est vérifiée, alors : $\exists\theta, \mathbf{C} = e^{i\theta}\mathbf{I}_d$.

Si deux matrices commutent, alors il existe une base qui les diagonalise simultanément (voir le Théorème 1.3.12 dans [HJ12]). Par conséquent, le fait que $\mathbf{C}$ commute avec tous les $\{\boldsymbol{x}_\ell \boldsymbol{x}_\ell^*\}_\ell$ implique que tous les $\{\boldsymbol{x}_\ell\}_\ell$ sont des vecteurs propres de $\mathbf{C}$ (car toute base qui diagonalise $\boldsymbol{x}_\ell \boldsymbol{x}_\ell^*$ contient $\boldsymbol{x}_\ell$ à une phase globale près). Appelons $e^{i\lambda_\ell}$ la valeur propre de $\mathbf{C}$ associée à $\boldsymbol{x}_\ell$ (elles est de module unitaire car $\mathbf{C}$ est unitaire).

Considérons deux indices ℓ_1 et ℓ_2, et montrons que $\boldsymbol{x}_{\ell_1} \not\perp \boldsymbol{x}_{\ell_2} \Rightarrow e^{i\lambda_{\ell_1}} = e^{i\lambda_{\ell_2}}$: $\boldsymbol{x}_{\ell_1}^* \boldsymbol{x}_{\ell_2} = (\mathbf{C}\boldsymbol{x}_{\ell_1})^* \mathbf{C}\boldsymbol{x}_{\ell_2} = e^{i(\lambda_{\ell_2} - \lambda_{\ell_1}} \boldsymbol{x}_{\ell_1}^* \boldsymbol{x}_{\ell_2}$. Si $\boldsymbol{x}_{\ell_1} \not\perp \boldsymbol{x}_{\ell_2}$,alors on peut diviser les deux cotés de l'équation par $\boldsymbol{x}_{\ell_1}^* \boldsymbol{x}_{\ell_2}$ et on a $1 = e^{i(\lambda_{\ell_2} - \lambda_{\ell_1})} \Rightarrow e^{i\lambda_{\ell_1}} = e^{i\lambda_{\ell_2}}$.

Soit ℓ_1 dans $\{1, ..., n_x\}$. $\boldsymbol{x}_{\ell_1}$ n'est orthogonale à aucune colonne de $\mathbf{F}_S(\boldsymbol{x}_{\ell_1})$ (par définition de $\mathbf{F}_S$), et les colonnes de $\mathbf{F}_S(\boldsymbol{x}_{\ell_1})$ sont également dans $\{\boldsymbol{x}_\ell\}_\ell$. Ainsi, toutes les colonnes de $\mathbf{F}_S(\boldsymbol{x}_{\ell_1})$ ont $e^{i\lambda_{\ell_1}}$ comme valeur propre associée. Il en va de même pour les colonnes de $\mathbf{F}_S^k(\boldsymbol{x}_{\ell_1})$ pour tout $k \geq 1$ (simple à montrer par récurrence). En particulier, toutes les colonnes de $\mathbf{F}_S^{n_x}(\boldsymbol{x}_{\ell_1})$ ont la même valeur propre associée : $e^{i\lambda_{\ell_1}}$. Mais (3.19) garantit que $\mathbf{F}_S^{n_x}(\boldsymbol{x}_{\ell_1})$ est de rang d. Il y a donc d colonnes linéairement indépendantes de $\mathbf{F}_S^{n_x}(\boldsymbol{x}_{\ell_1})$ qui forment une base, ce sont aussi des vecteurs propres de $\boldsymbol{C}$ (tous les $\{\boldsymbol{x}_\ell\}_\ell$ sont des vecteurs propres de $\mathbf{C}$). Cette base est donc une base propre de $\boldsymbol{C}$, et il n'y a qu'une seule valeur propre associée : $e^{i\lambda_{\ell_1}}$. Cela signifie que $\mathbf{C} = e^{i\lambda_{\ell_1}}\mathbf{I}_d$ cqfd.

B.4 Calcul des informations de Fisher

Dans cette annexe, nous montrons que les informations de Fisher pour la version exacte (multinomiale) et gaussienne de la vraisemblance sont les suivante :

$$\begin{aligned}\mathbf{I}^{exact}(\boldsymbol{\theta}_0) &= n_c \mathbf{J}(\boldsymbol{\theta}_0)^* \begin{pmatrix} p_1(\boldsymbol{\theta}_0) & & \\ & \ddots & \\ & & p_{n_{prob}}(\boldsymbol{\theta}_0) \end{pmatrix}^{-1} \mathbf{J}(\boldsymbol{\theta}_0) \\ \mathbf{I}^{gauss}(\boldsymbol{\theta}_0) &= n_c \mathbf{J}(\boldsymbol{\theta}_0)^* \begin{pmatrix} \widetilde{p}_1(\boldsymbol{\theta}_0) & & \\ & \ddots & \\ & & \widetilde{p}_{n_{prob}}(\boldsymbol{\theta}_0) \end{pmatrix}^{-1} \mathbf{J}(\boldsymbol{\theta}_0).\end{aligned} \tag{B.9}$$

Commençons par la vraisemblance gaussienne. Dans l'Annexe A.2.3, nous avons montré que la log-vraisemblance négative d'un écart aux probabilités théorique $\boldsymbol{\epsilon}$ s'écrivait $\mathcal{L}^{gauss}_{(\boldsymbol{x}',\boldsymbol{y}')}(\boldsymbol{p}) = n_c \sum_{\ell=1}^{n_{prob}} \frac{\varepsilon_\ell(\boldsymbol{p},\boldsymbol{x}',\boldsymbol{y}')^2}{\widetilde{p}_\ell}$. C'est la même vraisemblance (à une constante additive près) que pour un vecteur gaussien de taille n_{prob} centré de matrice de covariance diagonale dont le ℓ-ème élément vaut $\frac{\widetilde{p}_\ell}{n_c}$. Or l'information de Fisher pour un vecteur gaussien centré en $\boldsymbol{p}$ (les probabilités théoriques) dont la matrice de covariance $\boldsymbol{\Sigma}$ ne dépend par des paramètres à estimer vaut $J^*_{\boldsymbol{\epsilon}} \boldsymbol{\Sigma}^{-1} J_{\boldsymbol{\epsilon}}$ où $J_{\boldsymbol{\epsilon}}$ est la jacobienne de $\boldsymbol{p}$ en fonction des paramètres à estimer. On retrouve la formule à démontrer.

Pour la vraisemblance exacte, il nous faut faire le calcul. L'expression de l'information de Fisher est $\mathbf{I}_f(\boldsymbol{\theta}_0) = -\left(\mathbb{E}\left(\frac{\partial^2 \mathcal{L}}{\partial\theta_j\theta_k}\right)\right)_{j,k}$ où $\frac{\partial^2 .}{\partial\theta_j\theta_k}$ représente la dérivation successive par le j-ème puis le k-ème élement du vecteur $\boldsymbol{\theta}_0$. On rappelle l'expression de la log vraisemblance négative exacte : $\mathcal{L}^{exact}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = -\sum_{\ell=1}^{n_{prob}} n_\ell log(p_\ell(\boldsymbol{\theta}_0))$. où n_ℓ le ℓ-ème élément du vecteur des nombres d'occurrences observés $\boldsymbol{n}$ et $p_k(\boldsymbol{\theta}_0)$ est le k-ème élément du vecteur des probabilités théoriques $\boldsymbol{p}$ qui dépend des valeurs des paramètres $\boldsymbol{\theta}_0$, c'est le vecteur dont $\mathbf{J}(\boldsymbol{\theta}_0)$ est la jacobienne en fonction de $\boldsymbol{\theta}_0$. On a $\frac{\partial^2 \mathcal{L}}{\partial\theta_j\theta_k} = -\sum_{\ell=1}^{n_{prob}} n_\ell \frac{1}{p_\ell^2} \frac{\partial p_\ell}{\partial \boldsymbol{\theta}_j} \frac{\partial p_\ell}{\partial \boldsymbol{\theta}_k}$, si on prend l'espérance de cette quantité, on change n_k (le seul élément aléatoire) en $n_c p_k$ (l'espérance du nombre d'occurrence est la probabilité d'occurrence multipliée par le nombre de tirages) et on a $(\mathbf{I}^{exact}(\boldsymbol{\theta}_0))_{j,k} = \sum_{\ell=1}^{n_{prob}} \frac{n_c p_\ell}{p_\ell^2} \frac{\partial p_\ell}{\partial \boldsymbol{\theta}_j} \frac{\partial p_k}{\partial \boldsymbol{\theta}_k} = n_c \sum_{\ell=1}^{n_{prob}} \frac{\partial p_\ell}{\partial \boldsymbol{\theta}_j} \frac{1}{p_\ell} \frac{\partial p_\ell}{\partial \boldsymbol{\theta}_k}$, comme $\mathbf{J}(\boldsymbol{\theta}_0)$ est la jacobienne du vecteur des probabilités théoriques par rapport à $\boldsymbol{\theta}_0$, on retrouve la formule à démontrer.

B.5 Lien entre notre métrique et la fidélité

Dans la section 1.1.9, on définit la fidélité entre deux processus : $f(\epsilon, \widehat{\epsilon}) = tr\left(\sqrt{\sqrt{\boldsymbol{\rho}_\epsilon}\boldsymbol{\rho}_{\widehat{\epsilon}}\sqrt{\boldsymbol{\rho}_\epsilon}}\right)$ où $\boldsymbol{\rho}_\epsilon$ est l'état associé au processus par l'isomorphisme de Choi–Jamiołkowski, et pour un processus unitaire : $\epsilon(\boldsymbol{\rho}) = \mathbf{M}\boldsymbol{\rho}\mathbf{M}^*$. On rappelle l'expression de $\boldsymbol{\rho}_\epsilon$ dans (1.13)

$$\boldsymbol{\rho}_\epsilon = \frac{1}{d} \sum_{j,k=1}^{d} \mathbf{B}_{j,k} \otimes \epsilon(\mathbf{B}_{j,k}). \tag{B.10}$$

Où $\mathbf{B}_{j,k}$ est la matrice qui vaut zéro partout sauf en j,k où elle vaut 1. Pour un processus unitaire représenté par $\mathbf{M}$, $\boldsymbol{\rho}_\epsilon$ devient $\frac{1}{d}\sum_{j,k=1}^{d} \mathbf{B}_{j,k} \otimes (\mathbf{M}\mathbf{B}_{j,k}\mathbf{M}^*)$. Par définition du produit tensoriel, $\boldsymbol{\rho}_\epsilon$ est donc composé de d^2 bloc de $d \times d$ matrices, et le bloc en position j,k vaut $\mathbf{M}\mathbf{B}_{j,k}\mathbf{M}^* = \mathbf{M}\boldsymbol{\delta}_j(\mathbf{M}\boldsymbol{\delta}_k)^*$ où $\boldsymbol{\delta}_j$ est le vecteur de taille d qui vaut 1 en j et 0 ailleurs. Écrit ainsi, il est apparent que, si on définit le vecteur de taille d^2 $\boldsymbol{v}_{\mathbf{M}} = \frac{1}{\sqrt{d}} \begin{bmatrix} \mathbf{M}\boldsymbol{\delta}_1 \\ \vdots \\ \mathbf{M}\boldsymbol{\delta}_d \end{bmatrix}$, alors on a $\boldsymbol{\rho}_\epsilon = \boldsymbol{v}_{\mathbf{M}}\boldsymbol{v}^*_{\mathbf{M}}$

Comme les matrices $\boldsymbol{\rho}_\epsilon$ et $\boldsymbol{\rho}_{\widehat{\epsilon}}$ sont de rang 1, on peut simplifier l'expression de $f(\epsilon,\widehat{\epsilon}) = |\boldsymbol{v}^*_{\mathbf{M}}\boldsymbol{v}_{\widehat{\mathbf{M}}}|$ avec le même calcul que dans l'Annexe A.4. Développons $|\boldsymbol{v}^*_{\mathbf{M}}\boldsymbol{v}_{\widehat{\mathbf{M}}}|$

$$f(\epsilon,\widehat{\epsilon}) = |\boldsymbol{v}^*_{\mathbf{M}}\boldsymbol{v}_{\widehat{\mathbf{M}}}| = \left|\frac{1}{d}\begin{bmatrix}\mathbf{M}\boldsymbol{\delta}_1\\ \vdots \\ \mathbf{M}\boldsymbol{\delta}_d\end{bmatrix}^* \begin{bmatrix}\widehat{\mathbf{M}}\boldsymbol{\delta}_1\\ \vdots \\ \widehat{\mathbf{M}}\boldsymbol{\delta}_d\end{bmatrix}\right| = \frac{1}{d}\left|\sum_{j=1}^{d}\boldsymbol{\delta}_j^*\mathbf{M}^*\widehat{\mathbf{M}}\boldsymbol{\delta}_j\right| = \frac{1}{d}\left|tr(\mathbf{M}^*\widehat{\mathbf{M}})\right|$$

On rappelle l'expression de notre métrique : $\mu_p(\mathbf{M},\widehat{\mathbf{M}}) = \frac{1}{\sqrt{2d}}||\mathbf{M} - \widehat{\mathbf{M}}e^{i\phi}||$ où $\phi \in [0, 2\pi]$ est l'angle qui minimise la métrique. En fait, ce critère est très similaire à μ_s qui nous servait à définir l'erreur de QST. En effet, si on définit $\boldsymbol{m}$ (resp. $\widehat{\boldsymbol{m}}$) comme le vecteur de taille d^2 qui est la concaténation verticale de toutes les colonnes de $\mathbf{M}$ (resp. $\widehat{\mathbf{M}}$) multipliée par $\frac{1}{\sqrt{d}}$, alors, on a $\mu_p(\mathbf{M},\widehat{\mathbf{M}}) = \mu_s(\widehat{\mathbf{m}},\mathbf{m})$. Or, on a montré dans l'Annexe A.4 que $\mu_s^2(\boldsymbol{v},\widehat{\boldsymbol{v}}) = 1 - |\boldsymbol{v}^*\widehat{\boldsymbol{v}}|\forall \boldsymbol{v}\widehat{\boldsymbol{v}}$. Donc $\mu_p^2(\mathbf{M},\widehat{\mathbf{M}}) = 1 - |\boldsymbol{m}^*\widehat{\boldsymbol{m}}| = 1 - \left|\frac{1}{d}tr(\mathbf{M}^*\widehat{\mathbf{M}})\right| = 1 - f(\widehat{\mathbf{M}},\mathbf{M})$.

On a donc $f(\epsilon,\widehat{\epsilon}) = (1 - \mu_p(\mathbf{M},\widehat{\mathbf{M}})^2)$

B.6 Tableaux des résultats de mesures et de leurs espérances

TABLE B.1 : Mesures expérimentales sur l'ordinateur quantique

-	$\mathbf{M}\boldsymbol{v}_1$	$\mathbf{M}\boldsymbol{v}_2$	$\mathbf{M}\boldsymbol{v}_3$	$\mathbf{M}\boldsymbol{v}_4$	$\mathbf{M}^2\boldsymbol{v}_1$	$\mathbf{M}^2\boldsymbol{v}_2$	$\mathbf{M}^2\boldsymbol{v}_3$	$\mathbf{M}^2\boldsymbol{v}_4$
ZZ 00	243	139	123	58	249	128	123	75
ZZ 01	6	107	4	52	1	122	0	45
ZZ 10	0	1	1	74	0	0	125	71
ZZ 11	1	3	122	66	0	0	2	59
ZX 00	126	244	54	107	129	249	52	129
ZX 01	122	4	71	1	121	1	64	0
ZX 10	2	2	82	142	0	0	72	121
ZX 11	0	0	43	0	0	0	62	0
ZY 00	120	123	70	55	138	132	64	78
ZY 01	129	124	59	58	112	118	61	54
ZY 10	1	2	63	73	0	0	61	63
ZY 11	0	1	58	64	0	0	64	55
XX 00	63	127	118	248	69	123	125	248
XX 01	55	1	1	2	64	0	125	1
XX 10	65	122	4	0	57	127	0	1
XX 11	67	0	127	0	60	0	0	0
YY 00	54	61	5	59	61	66	59	68
YY 01	63	54	112	50	63	56	62	71
YY 10	72	62	127	72	56	74	62	58
YY 11	61	73	6	69	70	54	67	53

TABLE B.2 : Espérance du nombre de mesures avec des portes Hadamard et CNOT sont parfaites

-	$\mathbf{M}\boldsymbol{v}_1$	$\mathbf{M}\boldsymbol{v}_2$	$\mathbf{M}\boldsymbol{v}_3$	$\mathbf{M}\boldsymbol{v}_4$	$\mathbf{M}^2\boldsymbol{v}_1$	$\mathbf{M}^2\boldsymbol{v}_2$	$\mathbf{M}^2\boldsymbol{v}_3$	$\mathbf{M}^2\boldsymbol{v}_4$
ZZ 00	250	125	125	62.5	250	125	125	62.5
ZZ 01	0	125	0	62.5	0	125	0	62.5
ZZ 10	0	0	0	62.5	0	0	125	62.5
ZZ 11	0	0	125	62.5	0	0	0	62.5
ZX 00	125	250	62.5	125	125	250	62.5	125
ZX 01	125	0	62.5	0	125	0	62.5	0
ZX 10	0	0	62.5	125	0	0	62.5	125
ZX 11	0	0	62.5	0	0	0	62.5	0
ZY 00	125	125	62.5	62.5	125	125	62.5	62.5
ZY 01	125	125	62.5	62.5	125	125	62.5	62.5
ZY 10	0	0	62.5	62.5	0	0	62.5	62.5
ZY 11	0	0	62.5	62.5	0	0	62.5	62.5
XX 00	62.5	125	125	250	62.5	125	125	250
XX 01	62.5	0	0	0	62.5	0	125	0
XX 10	62.5	125	0	0	62.5	125	0	0
XX 11	62.5	0	125	0	62.5	0	0	0
YY 00	62.5	62.5	0	62.5	62.5	62.5	62.5	62.5
YY 01	62.5	62.5	125	62.5	62.5	62.5	62.5	62.5
YY 10	62.5	62.5	125	62.5	62.5	62.5	62.5	62.5
YY 11	62.5	62.5	0	62.5	62.5	62.5	62.5	62.5

TABLE B.3 : Espérance du nombre de mesures avec les valeurs des états initiaux et du processus estimées par ML

-	$\mathbf{M}\boldsymbol{v}_1$	$\mathbf{M}\boldsymbol{v}_2$	$\mathbf{M}\boldsymbol{v}_3$	$\mathbf{M}\boldsymbol{v}_4$	$\mathbf{M}^2\boldsymbol{v}_1$	$\mathbf{M}^2\boldsymbol{v}_2$	$\mathbf{M}^2\boldsymbol{v}_3$	$\mathbf{M}^2\boldsymbol{v}_4$
ZZ 00	248	137	120	67.2	249	133	125	66.9
ZZ 01	0.04	111	0.95	48.8	0.16	115	0.19	55.4
ZZ 10	0.25	0.47	1.40	72.2	0.19	0.13	124	68.1
ZZ 11	1.26	1.16	126	61.5	0.09	4e-03	0.14	59.4
ZX 00	125	247	51.6	115	130	249	59.0	122
ZX 01	123	0.69	70.1	0.75	118	0.72	66.2	0.38
ZX 10	1.24	1.48	77.2	133	0.02	0.04	64.0	127
ZX 11	0.26	0.16	50.9	0.33	0.26	0.09	60.5	0.15
ZY 00	127	123	66.3	59.3	122	134	66.1	64.9
ZY 01	121	124	55.5	56.7	127	115	59.2	57.4
ZY 10	1.03	0.47	66.2	71.0	0.08	0.07	58.4	64.7
ZY 11	0.47	1.17	61.8	62.8	0.20	0.07	66.1	62.8
XX 00	65.4	125	124	247	66.9	122	122	248
XX 01	58.1	0.66	2.10	0.95	64.6	0.41	126	0.48
XX 10	60.8	123	4.83	1.07	63.9	127	0.29	1.20
XX 11	65.6	0.19	119	0.13	54.4	0.40	0.09	0.05
YY 00	57.1	54.7	1.97	55.4	64.1	64.3	57.8	72.9
YY 01	59.9	51.3	114	55.6	62.6	57.9	63.5	69.2
YY 10	71.2	69.4	130	74.8	58.5	70.5	66.7	56.6
YY 11	61.6	74.4	2.66	64.0	64.7	57.1	61.8	51.0

Annexe C

Annexe du chapitre 5

Dans cette annexe, nous donnons les démonstrations de résultats du chapitre 5.

C.1 Lien entre notre métrique et la fidélité

Dans la section 1.1.9, on définit la fidélité entre deux processus : $f(\epsilon,\widehat{\epsilon}) = tr\left(\sqrt{\sqrt{\boldsymbol{\rho}_\epsilon}\boldsymbol{\rho}_{\widehat{\epsilon}}\sqrt{\boldsymbol{\rho}_\epsilon}}\right)$ où $\boldsymbol{\rho}_\epsilon$ est l'état associé au processus par l'isomorphisme de Choi–Jamiołkowski, et pour un processus unitaire : $\epsilon(\boldsymbol{\rho}) = \mathbf{M}\boldsymbol{\rho}\mathbf{M}^*$. On rappelle l'expression de $\boldsymbol{\rho}_\epsilon$ dans (1.13)

$$\boldsymbol{\rho}_\epsilon = \frac{1}{d}\sum_{j,k=1}^{d} \mathbf{M}_{j,k} \otimes \epsilon(\mathbf{M}_{j,k}). \tag{C.1}$$

Où $\mathbf{M}_{j,k}$ est la matrice qui vaut zéro partout sauf en j,k où elle vaut 1 (ne pas confondre avec $\mathbf{M}$ et $\widehat{\mathbf{M}}$). Pour un processus unitaire représenté par $\mathbf{M}$, $\boldsymbol{\rho}_\epsilon$ devient $\frac{1}{d}\sum_{j,k=1}^{d}\mathbf{M}_{j,k}\otimes(\mathbf{M}\mathbf{M}_{j,k}\mathbf{M}^*)$. Par définition du produit tensoriel, $\boldsymbol{\rho}_\epsilon$ est donc composé de d^2 bloc de $d\times d$ matrices, et le bloc en position j,k vaut $\mathbf{M}\mathbf{M}_{j,k}\mathbf{M}^* = \mathbf{M}\boldsymbol{\delta}_j(\mathbf{M}\boldsymbol{\delta}_k)^*$ où $\boldsymbol{\delta}_j$ est le vecteur de taille d qui vaut 1 en j et 0 ailleurs. Écrit ainsi, il est apparent que, si on définit le vecteur de taille d^2 $\boldsymbol{v}_\mathbf{M} = \frac{1}{\sqrt{d}}\begin{bmatrix}\mathbf{M}\boldsymbol{\delta}_1\\ \vdots \\ \mathbf{M}\boldsymbol{\delta}_d\end{bmatrix}$

$\boldsymbol{\rho}_\epsilon = \boldsymbol{v}_\mathbf{M}\boldsymbol{v}_\mathbf{M}^*$

Comme les matrices $\boldsymbol{\rho}_\epsilon$ et $\boldsymbol{\rho}_{\widehat{\epsilon}}$ sont de rang 1, on peut simplifier l'expression de $f(\epsilon,\widehat{\epsilon}) = |\boldsymbol{v}_\mathbf{M}^*\boldsymbol{v}_{\widehat{\mathbf{M}}}|$ avec le même calcul que dans l'Annexe A.4. Développons $|\boldsymbol{v}_\mathbf{M}^*\boldsymbol{v}_{\widehat{\mathbf{M}}}|$

$$f(\epsilon,\widehat{\epsilon}) = |\boldsymbol{v}_\mathbf{M}^*\boldsymbol{v}_{\widehat{\mathbf{M}}}| = \left|\frac{1}{d}\begin{bmatrix}\mathbf{M}\boldsymbol{\delta}_1\\ \vdots \\ \mathbf{M}\boldsymbol{\delta}_d\end{bmatrix}^*\begin{bmatrix}\widehat{\mathbf{M}}\boldsymbol{\delta}_1\\ \vdots \\ \widehat{\mathbf{M}}\boldsymbol{\delta}_d\end{bmatrix}\right| = \frac{1}{d}\left|\sum_{j=1}^{d}\boldsymbol{\delta}_j^*\mathbf{M}^*\widehat{\mathbf{M}}\boldsymbol{\delta}_j\right| = \frac{1}{d}\left|tr(\mathbf{M}^*\widehat{\mathbf{M}})\right|$$

On rappelle l'expression de notre métrique : $\mu_p(\mathbf{M},\widehat{\mathbf{M}}) = \frac{1}{\sqrt{2d}}||\mathbf{M}-\widehat{\mathbf{M}}e^{i\phi}||$ où $\phi\in[0,2\pi]$ est l'angle qui minimise la métrique. En fait, ce critère est très similaire à μ_s qui nous servait à définir l'erreur de QST. En effet, si on définit $\boldsymbol{m}$ (resp. $\widehat{\boldsymbol{m}}$) comme le vecteur de taille d^2 qui est la concaténation verticale de toutes les colonnes de $\mathbf{M}$ (resp. $\widehat{\mathbf{M}}$) multipliée par $\frac{1}{\sqrt{d}}$, alors, on a $\mu_p(\mathbf{M},\widehat{\mathbf{M}}) = \mu_s(\widehat{\mathbf{m}},\mathbf{m})$. Or, on a montré dans l'Annexe A.4 que $\mu_s^2(\boldsymbol{v},\widehat{\boldsymbol{v}}) = 1-|\boldsymbol{v}^*\widehat{\boldsymbol{v}}|\forall\boldsymbol{v}\widehat{\boldsymbol{v}}$. Donc $\mu_p^2(\mathbf{M},\widehat{\mathbf{M}}) = 1-|\boldsymbol{m}^*\widehat{\boldsymbol{m}}| = 1-\left|\frac{1}{d}tr(\mathbf{M}^*\widehat{\mathbf{M}})\right| = 1-f(\widehat{\mathbf{M}},\mathbf{M})$.

On a donc $f(\epsilon,\widehat{\epsilon}) = (1-\mu_p(\mathbf{M},\widehat{\mathbf{M}})^2)$

C.2 Tableaux des résultats de mesures et de leurs espérances

TABLE C.1 : Mesures expérimentales sur l'ordinateur quantique

-	$\mathbf{M}\boldsymbol{v}_1$	$\mathbf{M}\boldsymbol{v}_2$	$\mathbf{M}\boldsymbol{v}_3$	$\mathbf{M}\boldsymbol{v}_4$	$\mathbf{M}^2\boldsymbol{v}_1$	$\mathbf{M}^2\boldsymbol{v}_2$	$\mathbf{M}^2\boldsymbol{v}_3$	$\mathbf{M}^2\boldsymbol{v}_4$
ZZ 00	243	139	123	58	249	128	123	75
ZZ 01	6	107	4	52	1	122	0	45
ZZ 10	0	1	1	74	0	0	125	71
ZZ 11	1	3	122	66	0	0	2	59
ZX 00	126	244	54	107	129	249	52	129
ZX 01	122	4	71	1	121	1	64	0
ZX 10	2	2	82	142	0	0	72	121
ZX 11	0	0	43	0	0	0	62	0
ZY 00	120	123	70	55	138	132	64	78
ZY 01	129	124	59	58	112	118	61	54
ZY 10	1	2	63	73	0	0	61	63
ZY 11	0	1	58	64	0	0	64	55
XX 00	63	127	118	248	69	123	125	248
XX 01	55	1	1	2	64	0	125	1
XX 10	65	122	4	0	57	127	0	1
XX 11	67	0	127	0	60	0	0	0
YY 00	54	61	5	59	61	66	59	68
YY 01	63	54	112	50	63	56	62	71
YY 10	72	62	127	72	56	74	62	58
YY 11	61	73	6	69	70	54	67	53

TABLE C.2 : Espérance du nombre de mesures avec des portes Hadamard et CNOT sont parfaites

-	$\mathbf{M}\boldsymbol{v}_1$	$\mathbf{M}\boldsymbol{v}_2$	$\mathbf{M}\boldsymbol{v}_3$	$\mathbf{M}\boldsymbol{v}_4$	$\mathbf{M}^2\boldsymbol{v}_1$	$\mathbf{M}^2\boldsymbol{v}_2$	$\mathbf{M}^2\boldsymbol{v}_3$	$\mathbf{M}^2\boldsymbol{v}_4$
ZZ 00	250	125	125	62.5	250	125	125	62.5
ZZ 01	0	125	0	62.5	0	125	0	62.5
ZZ 10	0	0	0	62.5	0	0	125	62.5
ZZ 11	0	0	125	62.5	0	0	0	62.5
ZX 00	125	250	62.5	125	125	250	62.5	125
ZX 01	125	0	62.5	0	125	0	62.5	0
ZX 10	0	0	62.5	125	0	0	62.5	125
ZX 11	0	0	62.5	0	0	0	62.5	0
ZY 00	125	125	62.5	62.5	125	125	62.5	62.5
ZY 01	125	125	62.5	62.5	125	125	62.5	62.5
ZY 10	0	0	62.5	62.5	0	0	62.5	62.5
ZY 11	0	0	62.5	62.5	0	0	62.5	62.5
XX 00	62.5	125	125	250	62.5	125	125	250
XX 01	62.5	0	0	0	62.5	0	125	0
XX 10	62.5	125	0	0	62.5	125	0	0
XX 11	62.5	0	125	0	62.5	0	0	0
YY 00	62.5	62.5	0	62.5	62.5	62.5	62.5	62.5
YY 01	62.5	62.5	125	62.5	62.5	62.5	62.5	62.5
YY 10	62.5	62.5	125	62.5	62.5	62.5	62.5	62.5
YY 11	62.5	62.5	0	62.5	62.5	62.5	62.5	62.5

TABLE C.3 : Espérance du nombre de mesures avec les valeurs des états initiaux et du processus estimées par ML

-	$\mathbf{M}\boldsymbol{v}_1$	$\mathbf{M}\boldsymbol{v}_2$	$\mathbf{M}\boldsymbol{v}_3$	$\mathbf{M}\boldsymbol{v}_4$	$\mathbf{M}^2\boldsymbol{v}_1$	$\mathbf{M}^2\boldsymbol{v}_2$	$\mathbf{M}^2\boldsymbol{v}_3$	$\mathbf{M}^2\boldsymbol{v}_4$
ZZ 00	248	137	120	67.2	249	133	125	66.9
ZZ 01	0.04	111	0.95	48.8	0.16	115	0.19	55.4
ZZ 10	0.25	0.47	1.40	72.2	0.19	0.13	124	68.1
ZZ 11	1.26	1.16	126	61.5	0.09	4e-03	0.14	59.4
ZX 00	125	247	51.6	115	130	249	59.0	122
ZX 01	123	0.69	70.1	0.75	118	0.72	66.2	0.38
ZX 10	1.24	1.48	77.2	133	0.02	0.04	64.0	127
ZX 11	0.26	0.16	50.9	0.33	0.26	0.09	60.5	0.15
ZY 00	127	123	66.3	59.3	122	134	66.1	64.9
ZY 01	121	124	55.5	56.7	127	115	59.2	57.4
ZY 10	1.03	0.47	66.2	71.0	0.08	0.07	58.4	64.7
ZY 11	0.47	1.17	61.8	62.8	0.20	0.07	66.1	62.8
XX 00	65.4	125	124	247	66.9	122	122	248
XX 01	58.1	0.66	2.10	0.95	64.6	0.41	126	0.48
XX 10	60.8	123	4.83	1.07	63.9	127	0.29	1.20
XX 11	65.6	0.19	119	0.13	54.4	0.40	0.09	0.05
YY 00	57.1	54.7	1.97	55.4	64.1	64.3	57.8	72.9
YY 01	59.9	51.3	114	55.6	62.6	57.9	63.5	69.2
YY 10	71.2	69.4	130	74.8	58.5	70.5	66.7	56.6
YY 11	61.6	74.4	2.66	64.0	64.7	57.1	61.8	51.0

Table des figures

Liste des tableaux

Bibliographie

[AAB+19] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.* "Quantum supremacy using a programmable superconducting processor". *Nature*, volume 574, no. 7779, pp. 505–510, 2019. 11

[ABJ+03] J. B. Altepeter, D. Branning, E. Jeffrey, T. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White. "Ancilla-assisted quantum process tomography". *Physical Review Letters*, volume 90, no. 19, p. 193601, 2003. 21

[AFS22] S. T. Ahmad, A. Farooq, and H. Shin. "Self-guided quantum state tomography for limited resources". *Scientific Reports*, volume 12, no. 1, 2022. doi :10.1038/s41598-022-09143-7. 17

[Aru92] K. Arun. "A unitarily constrained total least squares problem in signal processing". *SIAM Journal on Matrix Analysis and Applications*, volume 13, no. 3, pp. 729–745, 1992. 71, 72

[Bal20] L. Ballentine. "Reviews of quantum foundations". *Physics Today*, volume 73, no. 6, pp. 11–12, 2020. 9

[BB14] C. H. Bennett and G. Brassard. "Quantum cryptography : Public key distribution and coin tossing". *Theoretical computer science*, volume 560, pp. 7–11, 2014. 6

[BCE06] R. Balan, P. Casazza, and D. Edidin. "On signal reconstruction without phase". *Applied and Computational Harmonic Analysis*, volume 20, no. 3, pp. 345–356, 2006. doi :10.1016/j.acha.2005.07.001. 18, 43

[BCMN14] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson. "Saving phase : Injectivity and stability for phase retrieval". *Applied and Computational Harmonic Analysis*, volume 37, no. 1, pp. 106–125, 2014. doi :10.1016/j.acha.2013.10.002. 18, 43

[BDK16] C. H. Baldwin, I. H. Deutsch, and A. Kalev. "Strictly-complete measurements for bounded-rank quantum-state tomography". *Physical Review A*, volume 93, no. 5, 2016. doi :10.1103/physreva.93.052105. 17, 20

[BDP+07] O. Bunk, A. Diaz, F. Pfeiffer, C. David, B. Schmitt, D. K. Satapathy, and J. F. Van Der Veen. "Diffractive imaging for periodic samples : retrieving one-dimensional concentration profiles across microfluidic channels". *Acta Crystallographica Section A : Foundations of Crystallography*, volume 63, no. 4, pp. 306–314, 2007. 18

[Ben80] P. Benioff. "The computer as a physical system : A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines". *Journal of statistical physics*, volume 22, no. 5, pp. 563–591, 1980. 6

[BK10] R. Blume-Kohout. "Hedged maximum likelihood quantum state estimation". *Physical Review Letters*, volume 105, no. 20, pp. 200504–200507, 2010. doi : 10.1103/physrevlett.105.200504. 137

[BKD14] C. H. Baldwin, A. Kalev, and I. H. Deutsch. "Quantum process tomography of unitary and near-unitary maps". *Phys Rev A*, volume 90, p. 012110, 2014. doi : 10.1103/PhysRevA.90.012110. 22, 24, 30, 32, 34, 70, 80, 104, 105, 107, 108, 109, 126, 127, 128, 158

[BKGN+13] R. Blume-Kohout, J. K. Gamble, E. Nielsen, J. Mizrahi, J. D. Sterk, and P. Maunz. "Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit". *arXiv preprint arXiv :13104492*, 2013. 27, 28

[BLSF19] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini. "Parameterized quantum circuits as machine learning models". *Quantum Science and Technology*, volume 4, no. 4, p. 043001, 2019. 29

[Bro70] C. G. Broyden. "The convergence of a class of double-rank minimization algorithms". *IMA Journal of Applied Mathematics*, volume 6, no. 1, pp. 76–90, 1970. doi :10.1093/imamat/6.1.76. 48, 51, 52, 85

[BWP+17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. "Quantum machine learning". *Nature*, volume 549, no. 7671, pp. 195–202, 2017. 6

[CDJ+13] J. Chen, H. Dawkins, Z. Ji, N. Johnston, D. Kribs, F. Shultz, and B. Zeng. "Uniqueness of quantum states compatible with given measurement results". *Physical Review A*, volume 88, no. 1, p. 012109, 2013. 7, 19, 20

[CFP16] R. J. Chapman, C. Ferrie, and A. Peruzzo. "Experimental demonstration of self-guided quantum tomography". *Physical Review Letters*, volume 117, no. 4, 2016. doi :10.1103/physrevlett.117.040402. 17

[CFYW19] Y. Chen, M. Farahzad, S. Yoo, and T.-C. Wei. "Detector tomography on ibm quantum computers and mitigation of an imperfect measurement". *Physical Review A*, volume 100, no. 5, p. 052315, 2019. 20, 37, 38

[Cho75] M.-D. Choi. "Completely positive linear maps on complex matrices". *Linear algebra and its applications*, volume 10, no. 3, pp. 285–290, 1975. 14, 15

[CJ10] P. Comon and C. Jutten. *Handbook of Blind Source Separation : Independent component analysis and applications*. Academic press, 2010. 29

[CKW+16] T. Cai, D. Kim, Y. Wang, M. Yuan, and H. H. Zhou. "Optimal large-scale quantum state tomography with pauli measurements". *The Annals of Statistics*, volume 44, no. 2, 2016. doi :10.1214/15-aos1382. 7, 17, 40

[CN97] I. L. Chuang and M. A. Nielsen. "Prescription for experimental determination of the dynamics of a quantum black box". *Journal of Modern Optics*, volume 44, no. 75, pp. 2455–2467, 1997. doi :10.1080/09500349708231894. 14, 16, 20, 21, 23, 33, 69, 90, 128

[CPF+10] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu. "Efficient quantum state tomography". *Nature communications*, volume 1, no. 1, p. 149, 2010. 19, 20

[CW20] J. Cotler and F. Wilczek. "Quantum overlapping tomography". *Physical Review Letters*, volume 124, no. 10, 2020. doi :10.1103/physrevlett.124.100401. 20, 38, 39, 40

[DD15] Y. Deville and A. Deville. "From blind quantum source separation to blind quantum process tomography". In *International Conference on Latent Variable Analysis and Signal Separation*, pp. 184–192. Springer, 2015. 29

[DD17a] Y. Deville and A. Deville. "Blind quantum source separation : Quantum-processing qubit uncoupling systems based on disentanglement". *Digital Signal Processing*, volume 67, pp. 30–51, 2017. 29, 30

[DD17b] Y. Deville and A. Deville. "The blind version of quantum process tomography : operating with unknown input values". *IFAC-PapersOnLine*, volume 50, no. 1, pp. 11731–11737, 2017. 29

[DD17c] Y. Deville and A. Deville. "Concepts and criteria for blind quantum source separation and blind quantum process tomography". *Entropy*, volume 19, no. 7, 2017. 29, 30

[DD20] Y. Deville and A. Deville. "Quantum process tomography with unknown single-preparation input states : Concepts and application to the qubit pair with internal exchange coupling". *Physical Review A*, volume 101, no. 4, p. 042332, 2020. 29, 30

[DDH+22] M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, and M. J. Reagor. "Entanglement perspective on the quantum approximate optimization algorithm". *Physical Review A*, volume 106, no. 2, p. 022423, 2022. 11, 38

[DiV00] D. P. DiVincenzo. "The physical implementation of quantum computation". *Fortschritte der Physik : Progress of Physics*, volume 48, no. 9-11, pp. 771–783, 2000. 1

[DLSS13] S. Das, K. Lochan, S. Sahu, and T. Singh. "Quantum to classical transition of inflationary perturbations : Continuous spontaneous localization as a possible mechanism". *Physical Review D*, volume 88, no. 8, p. 085020, 2013. 13

[DP01] G. D'Ariano and P. L. Presti. "Quantum tomography for measuring experimentally the matrix elements of an arbitrary quantum operation". *Physical Review Letters*, volume 86, no. 19, p. 4195, 2001. 21

[EAŻ05] J. Emerson, R. Alicki, and K. Życzkowski. "Scalable noise estimation with random unitary operators". *Journal of Optics B : Quantum and Semiclassical Optics*, volume 7, no. 10, p. S347, 2005. 28

[Fer14] C. Ferrie. "Self-guided quantum tomography". *Physical Review Letters*, volume 113, no. 19, 2014. doi :10.1103/physrevlett.113.190404. 17, 20, 41

[FGLE12] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. "Quantum tomography via compressed sensing : error bounds, sample complexity and efficient estimators". *New Journal of Physics*, volume 14, no. 9, p. 095022, 2012. 23, 24, 25

[Fin04] J. Finkelstein. "Pure-state informationally complete and "really" complete measurements". *Physical Review A*, volume 70, no. 5, 2004. doi :10.1103/physreva.70.052107. 18, 41

[Fiu01] J. Fiurášek. “Maximum-likelihood estimation of quantum measurement”. *Physical Review A*, volume 64, no. 2, p. 024102, 2001. 20

[FSC05] S. T. Flammia, A. Silberfarb, and C. M. Caves. “Minimal informationally complete measurements for pure states”. *Foundations of Physics*, volume 35, pp. 1985–2006, 2005. 19, 32, 33, 41

[GBMK13] T. M. Graham, J. T. Barreiro, M. Mohseni, and P. G. Kwiat. “Hyperentanglement-enabled direct characterization of quantum dynamics”. *Physical Review Letters*, volume 110, no. 6, p. 060404, 2013. 23

[GCE+15] D. Goyeneche, G. Cañas, S. Etcheverry, E. Gómez, G. Xavier, G. Lima, and A. Delgado. “Five measurement bases determine pure quantum states on any dimension”. *Physical Review Letters*, volume 115, no. 9, 2015. doi :10.1103/physrevlett.115.090401. 8, 19, 38, 41, 42, 45, 50, 63, 64, 156

[GCZ+19] M. Gong, M.-C. Chen, Y. Zheng, S. Wang, C. Zha, H. Deng, Z. Yan, H. Rong, Y. Wu, S. Li, *et al.* “Genuine 12-qubit entanglement on a superconducting quantum processor”. *Physical Review Letters*, volume 122, no. 11, p. 110501, 2019. 11

[GJ14] G. Gutoski and N. Johnston. “Process tomography for unitary quantum channels”. *Journal of Mathematical Physics*, volume 55, no. 3, p. 032201, 2014. 32

[GKKT20] M. Guţă, J. Kahn, R. Kueng, and J. A. Tropp. “Fast state tomography with optimal error bounds”. *Journal of Physics A : Mathematical and Theoretical*, volume 53, no. 20, p. 204001, 2020. 17

[GLF+10] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. “Quantum state tomography via compressed sensing”. *Physical Review Letters*, volume 105, no. 15, 2010. doi :10.1103/physrevlett.105.150401. 17, 40, 45, 50

[Gro96] L. K. Grover. “A fast quantum mechanical algorithm for database search”. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219. 1996. 6

[Gut09] A. Gut. *An Intermediate Course in Probability*. Springer, 2009. 137

[GVL13] G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU press, 2013. 72

[HHL09] A. W. Harrow, A. Hassidim, and S. Lloyd. “Quantum algorithm for linear systems of equations”. *Physical Review Letters*, volume 103, no. 15, p. 150502, 2009. 6

[HJ12] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012. 146

[HMW13] T. Heinosaari, L. Mazzarella, and M. M. Wolf. “Quantum tomography under prior information”. *Communications in Mathematical Physics*, volume 318, no. 2, pp. 355–374, 2013. doi :10.1007/s00220-013-1671-8. 18, 19, 43

[HŘFJ04] Z. Hradil, J. Řeháček, J. Fiurášek, and M. Ježek. “Three maximum-likelihood methods in quantum mechanics”. In *Quantum State Estimation*, pp. 59–112. Springer Berlin Heidelberg, 2004. doi :10.1007/978-3-540-44481-7_3. 51, 65

[HTF+20] Z. Hou, J.-F. Tang, C. Ferrie, G.-Y. Xiang, C.-F. Li, and G.-C. Guo. “Experimental realization of self-guided quantum process tomography”. *Physical Review A*, volume 101, no. 2, p. 022317, 2020. 25, 26, 27

[HWFZ20] H.-L. Huang, D. Wu, D. Fan, and X. Zhu. "Superconducting quantum computing : a review". *Science China Information Sciences*, volume 63, pp. 1–32, 2020. 7

[IB05] S. Imre and F. Balazs. *Quantum Computing and Communications : an engineering approach*. John Wiley & Sons, 2005. 6

[JFH03] M. Ježek, J. Fiurášek, and Z. Hradil. "Quantum inference of states and processes". *Physical Review A*, volume 68, no. 1, p. 012305, 2003. 21, 65

[JKMW01] D. F. James, P. G. Kwiat, W. J. Munro, and A. G. White. "Measurement of qubits". *Physical Review A*, volume 64, no. 5, p. 052312, 2001. 17

[JP85] J.-N. Juang and R. S. Pappa. "An eigensystem realization algorithm for modal parameter identification and model reduction". *Journal of guidance, control, and dynamics*, volume 8, no. 5, pp. 620–627, 1985. 33

[KBGK18] A. C. Keith, C. H. Baldwin, S. Glancy, and E. Knill. "Joint quantum-state and measurement tomography with incomplete measurements". *Physical Review A*, volume 98, no. 4, p. 042318, 2018. 20, 34, 68

[KK09] A. Kofman and A. Korotkov. "Two-qubit decoherence mechanisms revealed via quantum process tomography". *Physical Review A*, volume 80, no. 4, p. 042103, 2009. 25

[KKD15] A. Kalev, R. L. Kosut, and I. H. Deutsch. "Quantum tomography protocols with positivity are compressed sensing protocols". *npj Quantum Information*, volume 1, no. 1, 2015. doi :10.1038/npjqi.2015.18. 17, 40

[KLR+08] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. "Randomized benchmarking of quantum gates". *Physical Review A*, volume 77, no. 1, p. 012307, 2008. 28

[KLY15] S. Kimmel, G. H. Low, and T. J. Yoder. "Robust calibration of a universal single-qubit gate set via robust phase estimation". *Physical Review A*, volume 92, no. 6, p. 062315, 2015. 34, 70

[KMN+07] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn. "Linear optical quantum computing with photonic qubits". *Reviews of modern physics*, volume 79, no. 1, p. 135, 2007. 7

[KMW02] D. Kielpinski, C. Monroe, and D. J. Wineland. "Architecture for a large-scale ion-trap quantum computer". *Nature*, volume 417, no. 6890, pp. 709–711, 2002. 7

[Kra71] K. Kraus. "General state changes in quantum theory". *Annals of Physics*, volume 64, no. 2, pp. 311–335, 1971. 14, 15

[KTA+20] Y. Kim, Y. S. Teo, D. Ahn, D.-G. Im, Y.-W. Cho, G. Leuchs, L. L. Sánchez-Soto, H. Jeong, and Y.-H. Kim. "Universal compressive characterization of quantum dynamics". *Physical Review Letters*, volume 124, no. 21, p. 210401, 2020. 24

[LFCR+09] J. S. Lundeen, A. Feito, H. Coldenstrodt-Ronge, K. L. Pregnell, C. Silberhorn, T. C. Ralph, J. Eisert, M. B. Plenio, and I. A. Walmsley. "Tomography of quantum detectors". *Nature Physics*, volume 5, no. 1, pp. 27–30, 2009. 20

[LGDM23] I. López Gutiérrez, F. Dietrich, and C. B. Mendl. “Quantum process tomography of unitary maps from time-delayed measurements”. *Quantum Information Processing*, volume 22, no. 6, p. 251, 2023. 34

[LSS99] A. Luis and L. L. Sánchez-Soto. “Complete characterization of arbitrary quantum measurement processes”. *Physical Review Letters*, volume 83, no. 18, p. 3573, 1999. 20

[MGE11] E. Magesan, J. M. Gambetta, and J. Emerson. “Scalable and robust randomized benchmarking of quantum processes”. *Physical Review Letters*, volume 106, no. 18, p. 180504, 2011. 28

[MGE12] E. Magesan, J. M. Gambetta, and J. Emerson. “Characterizing quantum gates via randomized benchmarking”. *Physical Review A*, volume 85, no. 4, p. 042311, 2012. 28, 29

[MGS+13] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen. “Self-consistent quantum process tomography”. *Physical Review A*, volume 87, no. 6, p. 062119, 2013. 27

[MISE08] J. Miao, T. Ishikawa, Q. Shen, and T. Earnest. “Extending x-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes”. *Annu Rev Phys Chem*, volume 59, pp. 387–410, 2008. 18

[MJZ+16] X. Ma, T. Jackson, H. Zhou, J. Chen, D. Lu, M. D. Mazurek, K. A. G. Fisher, X. Peng, D. Kribs, K. J. Resch, Z. Ji, B. Zeng, and R. Laflamme. “Pure-state tomography with the expectation value of pauli operators”. *Physical Review A*, volume 93, no. 3, 2016. doi :10.1103/physreva.93.032140. 7, 17, 19, 20, 40

[ML06] M. Mohseni and D. Lidar. “Direct characterization of quantum dynamics”. *Physical Review Letters*, volume 97, no. 17, p. 170501, 2006. 23

[MN08] S. Morita and H. Nishimori. “Mathematical foundation of quantum annealing”. *Journal of Mathematical Physics*, volume 49, no. 12, p. 125210, 2008. 6

[MR09] M. Mohseni and A. Rezakhani. “Equation of motion for the process matrix : Hamiltonian identification and dynamical control of open quantum systems”. *Physical Review A*, volume 80, no. 1, p. 010101, 2009. 25

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000. 9, 10, 12, 13, 14, 17, 21, 39, 41, 42, 53

[NGR+21a] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout. “Gate set tomography”. *Quantum*, volume 5, p. 557, 2021. 27

[NGR+21b] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout. “Gate set tomography”. *Quantum*, volume 5, p. 557, 2021. doi :10.22331/q-2021-10-05-557. 27

[OSB15a] S. Omkar, R. Srikanth, and S. Banerjee. “Characterization of quantum dynamics using quantum error correction”. *Physical Review A*, volume 91, no. 1, p. 012324, 2015. 29

[OSB15b] S. Omkar, R. Srikanth, and S. Banerjee. “Quantum code for quantum error characterization”. *Physical Review A*, volume 91, no. 5, p. 052309, 2015. 29

[OSS+21] C. Outeiral, M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin, and C. M. Deane. "The prospects of quantum computing in computational molecular biology". *Wiley Interdisciplinary Reviews : Computational Molecular Science*, volume 11, no. 1, p. e1481, 2021. 6

[OWE19] E. Onorati, A. Werner, and J. Eisert. "Randomized benchmarking for individual quantum gates". *Physical Review Letters*, volume 123, no. 6, p. 060501, 2019. 28

[PCZ97] J. Poyatos, J. I. Cirac, and P. Zoller. "Complete characterization of a quantum process : the two-bit quantum gate". *Physical Review Letters*, volume 78, no. 2, p. 390, 1997. 20, 21

[Pea00] K. Pearson. "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, volume 50, no. 302, pp. 157–175, 1900. doi :10.1080/14786440009463897. 138

[Phi23] F. Phillipson. "Quantum computing in telecommunication—a survey". *Mathematics*, volume 11, no. 15, p. 3423, 2023. 6

[PKB+20] A. M. Palmieri, E. Kovlakov, F. Bianchi, D. Yudin, S. Straupe, J. D. Biamonte, and S. Kulik. "Experimental neural network enhanced quantum tomography". *npj Quantum Information*, volume 6, no. 1, p. 20, 2020. 29

[RGK13] D. M. Reich, G. Gualdi, and C. P. Koch. "Minimum number of input states required for quantum gate characterization". *Physical Review A*, volume 88, no. 4, p. 042309, 2013. 32, 80, 81, 127, 144

[RSW02] M. B. Ruskai, S. Szarek, and E. Werner. "An analysis of completely-positive trace-preserving maps on m2". *Linear Algebra and its Applications*, volume 347, no. 1-3, pp. 159–187, 2002. 14, 15

[RVB+14] A. V. Rodionov, A. Veitia, R. Barends, J. Kelly, D. Sank, J. Wenner, J. M. Martinis, R. L. Kosut, and A. N. Korotkov. "Compressed sensing quantum process tomography for superconducting quantum gates". *Physical Review B*, volume 90, no. 14, p. 144504, 2014. 23, 24

[Sho87] N. Z. Shor. "Quadratic optimization problems". *Soviet Journal of Computer and Systems Sciences*, volume 25, pp. 1–11, 1987. 44

[Sho94] P. W. Shor. "Algorithms for quantum computation : discrete logarithms and factoring". In *Proceedings 35th annual symposium on foundations of computer science*, pp. 124–134. Ieee, 1994. 6

[SKD+17] A. Shchukina, P. Kasprzak, R. Dass, M. Nowakowski, and K. Kazimierczuk. "Pitfalls in compressed sensing reconstruction and how to avoid them". *Journal of biomolecular NMR*, volume 68, pp. 79–98, 2017. 24

[SKM+11] A. Shabani, R. Kosut, M. Mohseni, H. Rabitz, M. Broome, M. Almeida, A. Fedrizzi, and A. White. "Efficient measurement of quantum dynamics via compressive sensing". *Physical Review Letters*, volume 106, no. 10, p. 100401, 2011. 23, 24, 25

[SM14] A. Shukla and T. Mahesh. "Single-scan quantum process tomography". *Physical Review A*, volume 90, no. 5, p. 052301, 2014. 21

[SRA+13] A. Smith, C. A. Riofrío, B. E. Anderson, H. Sosa-Martinez, I. H. Deutsch, and P. S. Jessen. "Quantum state tomography by continuous measurement and compressed sensing". *Physical Review A*, volume 87, no. 3, 2013. doi :10.1103/physreva.87.030102. 17, 40

[SSKKG22] T. Surawy-Stepney, J. Kahn, R. Kueng, and M. Guta. "Projected least-squares quantum process tomography". *Quantum*, volume 6, p. 844, 2022. 21, 38

[TEŘH11] Y. S. Teo, B.-G. Englert, J. Řeháček, and Z. Hradil. "Adaptive schemes for incomplete quantum process tomography". *Physical Review A*, volume 84, no. 6, p. 062125, 2011. 29

[TSK+20] Y. S. Teo, G. Struchalin, E. Kovlakov, D. Ahn, H. Jeong, S. Straupe, S. Kulik, G. Leuchs, and L. L. Sánchez-Soto. "Objective compressive quantum process tomography". *Physical Review A*, volume 101, no. 2, p. 022334, 2020. 23, 24

[VD22] F. Verdeil and Y. Deville. "Two unitary quantum process tomography algorithms robust to systematic errors". In *Physical Sciences Forum*, volume 5, p. 29. Multidisciplinary Digital Publishing Institute, 2022. 68

[VD23a] F. Verdeil and Y. Deville. "Pure-state tomography with parallel unentangled measurements". *Physical Review A*, volume 107, no. 1, p. 012408, 2023. 35

[VD23b] F. m. c. Verdeil and Y. Deville. "Unitary quantum process tomography with unreliable pure input states". *Phys Rev A*, volume 108, p. 062410, 2023. doi : 10.1103/PhysRevA.108.062410. 67

[VDD21] F. Verdeil, Y. Deville, and A. Deville. "Two-qubit unitary quantum process tomography by multiple-delay output measurements for one unknown input pure state value". In *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 161–165. IEEE, Rio de Janeiro, Brazil, 2021. doi :10.1109/SSP49050.2021.9513830. 68, 69

[VE19] L. M. Vandersypen and M. A. Eriksson. "Quantum computing with semiconductor spins". *Physics Today*, volume 72, no. 8, pp. 38–45, 2019. 7

[Vik06] T. Viklands. "Algorithms for the weighted orthogonal procrustes problem and other least squares problems". Ph.D. thesis, Datavetenskap, 2006. 130

[Wan13] Y. Wang. "Asymptotic equivalence of quantum state tomography and noisy matrix completion". *The Annals of Statistics*, volume 41, no. 5, 2013. doi : 10.1214/13-aos1156. 17, 40

[WBC+21] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, *et al.* "Strong quantum computational advantage using a superconducting quantum processor". *Physical Review Letters*, volume 127, no. 18, p. 180501, 2021. 11, 38

[WdM13] I. Waldspurger, A. d'Aspremont, and S. Mallat. "Phase recovery, MaxCut and complex semidefinite programming". *Mathematical Programming*, volume 149, no. 1-2, pp. 47–81, 2013. doi :10.1007/s10107-013-0738-9. 18, 19, 44, 45

[WDQ+17] Y. Wang, D. Dong, B. Qi, J. Zhang, I. R. Petersen, and H. Yonezawa. "A quantum hamiltonian identification algorithm : Computational complexity and error analysis". *IEEE Transactions on Automatic Control*, volume 63, no. 5, pp. 1388–1403, 2017. 33

[WHSK20] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais. "Qudits and high-dimensional quantum computing". *Frontiers in Physics*, volume 8, p. 589504, 2020. 7

[WZH+07] Z.-W. Wang, Y.-S. Zhang, Y.-F. Huang, X.-F. Ren, and G.-C. Guo. "Experimental realization of direct characterization of quantum dynamics". *Physical Review A*, volume 75, no. 4, p. 044304, 2007. 23

[XCZ+18] K. Xu, J.-J. Chen, Y. Zeng, Y.-R. Zhang, C. Song, W. Liu, Q. Guo, P. Zhang, D. Xu, H. Deng, *et al.* "Emulating many-body localization with a superconducting quantum processor". *Physical Review Letters*, volume 120, no. 5, p. 050507, 2018. 11, 38

[XLW+22] S. Xue, Y. Liu, Y. Wang, P. Zhu, C. Guo, and J. Wu. "Variational quantum process tomography of unitaries". *Physical Review A*, volume 105, no. 3, p. 032427, 2022. 29

[XNK+20] T. Xin, X. Nie, X. Kong, J. Wen, D. Lu, and J. Li. "Quantum pure state tomography via variational hybrid quantum-classical method". *Physical Review Applied*, volume 13, no. 2, p. 024013, 2020. 20

[Yat34] F. Yates. "Contingency tables involving small numbers and the χ 2 test". *Supplement to the Journal of the Royal Statistical Society*, volume 1, no. 2, pp. 217–235, 1934. 138

[ZS14] J. Zhang and M. Sarovar. "Quantum hamiltonian identification from measurement time traces". *Physical Review Letters*, volume 113, no. 8, p. 080401, 2014. 33

[Zur03] W. H. Zurek. "Decoherence, einselection, and the quantum origins of the classical". *Reviews of Modern Physics*, volume 75, no. 3, p. 715, 2003. 9

Résumé

L'objectif principal de cette thèse est de développer des algorithmes de tomographie de processus quantiques. Ce problème est étudié depuis la fin des années 1990 dans la littérature car les portes quantiques sont les blocs de base de la plupart des ordinateurs quantiques, et estimer leurs paramètres est nécessaire pour réaliser des portes de meilleure qualité. Le nombre de paramètres réels d'un processus quelconque est $2^{4n_{qb}} - 2^{2n_{qb}}$ (n_{qb} est le nombre de bits quantiques ou qubits), ce nombre devient très rapidement prohibitif (240 paramètres pour deux qubits, 4032 paramètres pour trois qubits). Afin de s'affranchir de ce problème, nous supposons que le processus à étudier est unitaire, ce qui est garanti dans un système fermé. Le nombre de paramètres d'un système unitaire dépend toujours exponentiellement du nombre de qubits mais de façon plus raisonnable : $2^{2n_{qb}}$ (16 pour deux qubits, 64 pour trois qubits). La méthode de tomographie de processus que nous proposons fonctionne avec des états d'entrée quelconques (ou presque), on suppose seulement qu'il est possible de préparer plusieurs copies d'un ensemble initialement inconnu d'états purs d'entrée et d'en mesurer avant et après que le processus à estimer ait été appliqué. Après avoir estimé les états et levé des indéterminations sur des paramètres à estimer, cet algorithme se résume à un problème de moindres carrés linéaires avec contrainte d'unitarité. Ce problème peut être résolu analytiquement et sans point initial, il est donc possible d'identifier une porte sans aucune connaissance préalable. Pour avoir une estimation plus précise des paramètres de la porte, un algorithme de maximum de vraisemblance (plus lent et nécessitant un bon point initial fourni par la version de base de l'algorithme) est proposé.

Nos algorithmes de tomographie de processus fonctionnent avec tout ensemble de types de mesures qui permet d'identifier les états, mais nous proposons nos types de mesures et algorithmes d'estimation d'états adaptés. Nous proposons aussi un algorithme qui permet d'identifier certains paramètres des mesures quantiques que nous utilisons et de faire la tomographie de processus en une seule expérience. Ainsi, nous pouvons envisager d'identifier un processus avec une précision qui ne dépendra ni de la précision avec laquelle on prépare des états d'entrée de référence (car notre algorithme fonctionne avec des états quelconques, et que tous les états qui nous intéressent sont mesurés et estimés), ni de la connaissance a priori que l'on a sur les mesures quantiques réalisées.

Mots clés : Tomographie de processus quantique, Tomographie d'état quantique, Tomographie de mesure quantique, Récupération de phase, Problème de Procrustes, Maximum de Vraisemblance

Abstract

The main objective of this thesis is to develop quantum process tomography algorithms. This problem has been studied since the late 1990s in the literature because quantum gates are the building blocks of most quantum computers, and estimating their parameters is necessary to make better gates. The number of real parameters for any process is $2^{4n_{qb}} - 2^{2n_{qb}}$ (n_{qb} is the number of quantum bits or qubits), this number quickly becomes prohibitive (240 parameters for two qubits, 4032 parameters for three qubits). To avoid this problem, we assume that the process to be studied is unitary, which is always the case in a closed system. The number of parameters for a unitary process still depends exponentially on the number of qubits, but more reasonably: $2^{2n_{qb}}$ (16 for two qubits, 64 for three qubits). The process tomography method we propose works with almost any input states, we just assume that it is possible to prepare several copies of an initially unknown set of pure input states and measure them before and after the process to be estimated has been applied. From this system, a process tomography algorithm based on pure state tomography is proposed. Once indeterminacies have been lifted, this algorithm boils down to a linear least squares problem with a unitarity constraint. This problem can be solved analytically and without an initial point, so it is possible to identify a gate without any prior knowledge. To get a more accurate estimate of the gate parameters, a maximum likelihood algorithm (which is slower and requires a good initial point provided by the basic version of the algorithm) is proposed.

Our process tomography algorithms work with any set of measurement types that identifies the states, but we propose our own measurement types and state estimation algorithms. We also propose an algorithm that can identify some of the parameters of the quantum measurements we use and perform process tomography with a single experiment. Therefore, we can hope to identify a process with an accuracy that will depend neither on the precision with which we prepare reference input states (since our algorithm works with any states, and all the states of interest are measured and estimated), nor on the prior knowledge we have of the quantum measurements we perform.

Keywords : Quantum process tomography, Quantum state tomography, Quantum measurement tomography, Phase recovery, Procrustes problem, Maximum likelihood