



Unitary quantum process tomography with unreliable pure input states

François Verdeil  and Yannick Deville 

Université de Toulouse, UPS, CNRS, CNES, OMP, IRAP, 31400 Toulouse, France



(Received 16 June 2023; accepted 24 November 2023; published 11 December 2023)

Quantum process tomography (QPT) methods aim at identifying a given quantum process. QPT is a major quantum information processing tool, since it allows one to characterize the actual behavior of quantum gates, which are the building blocks of quantum computers. The present paper focuses on the estimation of a unitary process. This class is of particular interest because quantum mechanics postulates that the evolution of any closed quantum system is described by a unitary transformation. The standard approach of QPT is to measure copies of a particular set of predetermined (generally pure) states after they have been modified by the process to be identified. The main problem with this setup is that preparing an input state and setting it precisely to a predetermined value is challenging and thus yields errors. These errors can be decomposed into a sum of centered errors (i.e., whose average on all the copies is zero) and systematic errors that are the same for all the copies. The latter is often the main source of error in QPT. The algorithm we introduce works for any input states that make QPT theoretically possible (i.e., unless there are several solutions due to, e.g., a lack of diversity in the input states). The fact that we do not require the input states to be precisely set to predetermined values means that we can use a trick to remove the issue of systematic errors by considering that some states' copies are unknown but measured before they go through the process to be identified. We achieve this by splitting the copies of each input state into several groups and measuring the copies of the k th group after they have successively been transferred through k instances of the process to be identified (each copy of each input state is used for only a single measurement). Using this approach, we can compute estimates of the measured states before and after they go through the process without using the knowledge we might have about the initial states. We test our algorithm with simulated data, and we assess its performance with a CNOT gate on a trapped-ions qubit quantum computer.

DOI: [10.1103/PhysRevA.108.062410](https://doi.org/10.1103/PhysRevA.108.062410)

I. INTRODUCTION

Quantum process tomography aims at identifying the quantum process associated with a physical quantum gate. It was first introduced in Refs. [1] and [2]; the former came up with the name quantum process tomography (QPT). They use copies of a set of known input states that are transformed by the process. Those transformed states are then measured and estimated using quantum state tomography (QST aims at estimating a quantum state using measurements). This method scales poorly when the number of qubits increases. This is unsurprising because, in general, a quantum process has $d^4 - d^2$ independent real parameters, with d the dimension of the Hilbert space (for an n_{qb} -qubit system $d = 2^{n_{qb}}$, see [3], p. 391).

This method and others inspired by it would later be called standard QPT (SQPT), in contrast to nonstandard QPT that uses ancilla qubits and weak measurements (see [4] for a survey). Shabani *et al.* [5] proposed a method that follows the SQPT approach and scales better with the number of qubits by assuming that the process matrix is sparse. This approach is very popular [6–8], but we chose not to use it for the following reason. Almost all useful processes have unitary target values (they are not unitary in practice because the implementation is not perfect but they are close to their target), and we think that, in general, assuming it is actually unitary (or equivalently that the rank of its process matrix is 1) is a better regularization

hypothesis than assuming that its process matrix is sparse. A rank 2 process matrix with two similar eigenvalues can be considered sparse, but the associated process is not a good approximation for a process that represents a unitary gate.

Thus, like Refs. [9–11], we choose to restrict ourselves to unitary processes. This class is of particular interest because the evolution of any closed quantum system is described by a unitary transformation. Baldwin *et al.* [9] study the tomography of unitary processes (and near-unitary ones). They use the work of Reich *et al.* [10] to choose their input states. Reich *et al.* studied the tomography of unitary processes; they established a necessary and sufficient condition to be able to distinguish a unitary process from any other process (unitary or not), but did not explicitly propose a QPT algorithm. In [11] the authors also study unitary processes. Their algorithm is adapted to a universal single-qubit gate set. A major advantage of the algorithm in [11] is that it is robust to faulty input states and faulty measurements. In our opinion, its main drawback is that it is defined only for single-qubit gates, which need QPT the least: Keith *et al.* [12] remark that “single-qubit gates have been demonstrated with high fidelity, but entangling gates have lower fidelities.”

Unitary processes are easier to handle partly because they involve “only” d^2 independent real parameters ($d^2 - 1$ without the global phase). This number still scales exponentially with the number of qubits, but is more reasonable than the $d^4 - d^2$ real parameters that [1] identifies. The method of [1]

is experimentally realistic for only one or two qubits, and even with two qubits there are $4^4 - 4^2 = 240$ real parameters. This lower number of parameters to be estimated for a unitary process means that QPT can be performed with fewer input states (d input states in Refs. [9] and [11] vs d^2 input states for the SQPT of [1]). Another advantage of unitary processes is that they are relatively simple to parametrize. A unitary process is uniquely parameterized by a unitary matrix (the matrix \mathbf{M} defined in Sec. II A, to be distinguished from the above-mentioned process matrix), up to a global phase. In contrast, the Kraus-operator representation of nonunitary processes is highly nonunique [13], and assessing the differences between two Kraus-operator representations (target gate and estimated gate, for example) is not as straightforward as comparing two unitary matrices up to a global phase. Unitary processes also have the advantage of preserving the purity of the input states; therefore unitary QPT generally assumes that input and output states are pure. This is an advantage because pure states have fewer parameters than mixed states and require fewer types of measurements to be identified (see [14]).

Beyond the number of parameters of quantum processes, a major issue with algorithms that use the SQPT setup is the fact that the input states have to be known, and, in the case of the unitary QPT algorithm in [9], the values of the known input states are imposed by the method (this is what we mean by “predetermined”). This is problematic because, to prepare a quantum state, one generally has to use a quantum gate, and to know with a high degree of accuracy which state is prepared, one must know the gate with a high degree of accuracy. This would require QPT. We are not the first to point out and to try to get around this problem. Reference [15] first identified this problem and tried to solve it by designing an algorithm that simultaneously identifies all the gates involved in the setup (including the gates used for state preparation) with no unitarity assumption. Later, gate set tomography (GST) was introduced [15,16]. It goes a step further, by simultaneously identifying the initial state, the processes applied to it, and the types of measurements performed. The identification is possible only up to significant indeterminacies (called a gauge) and involves many parameters that make GST very difficult when d increases.

We also proposed our own solution to this problem: In 2015 we introduced the blind version of QPT (BQPT) [17], then detailed it in [18] and more recently in [19]. In those papers we focused on the tomography of the two-qubit cylindrical-symmetry Heisenberg coupling process. For those algorithms, the operator has to prepare one or several copies of an unknown (hence the “blind”) set of initial states [20]. This removes the issue of systematic errors (with respect to a desired state) during the preparation. The system is identified by processing output measurements associated with n_i different unknown input states going through the system. Generally, we need to perform QST or at least to estimate some measurement outcome probabilities for each of the n_i output states. For the approaches of [17,18], this kind of QST requires n_c copies of each considered output state. Therefore for each one of the n_i states the same experiment has to be repeated n_c times with the same input state value, thus yielding $n_i \times n_c$ input state preparations in total. A more recent paper [19] also proposes “single-preparation BQPT methods,” i.e., methods

which can operate with only one instance of each considered input state, i.e., $n_c = 1$. Finally, in Refs. [21] and [22] we introduced semiblind setups that are generalized by the setup that is presented in the present paper. The idea is to split the initial states copies into n_s ($n_s = d$ for [21] and $n_s = 2$ for [22]) groups, and measure the copies of the k th group, after they have gone through the process k (or $k - 1$ for [22]) times. Thanks to this trick, we can use a QST algorithm to identify the states before and after they go through the process. Therefore, we can estimate the unitary process without having to trust the values of the states it is applied to because they are estimated via QST. We use the term “semiblind” because the constraints on the input states are very loose, but some copies of some states are measured before they are transformed by the process to be identified (each copy is measured only once).

In the current paper, we introduce a unitary QPT algorithm that uses nonpredetermined input states; this is a step beyond [9]. We also provide a necessary and sufficient condition (like in [10]) on the measured states that guarantees that QPT is possible (i.e., that \mathbf{M} can be identified up to a global phase). This condition happens to be very loose, and it can be met with weakly constrained input states. Furthermore we succeeded in proving that our QPT algorithm works whenever this constraint is satisfied. This is remarkable as it guarantees that when we are unable to perform QPT with our algorithm, it is because QPT is impossible for any QPT algorithm.

Our QPT algorithm could work with an SQPT setup, i.e., with predetermined input state values. But since it works with (almost) any set of values, we chose to use the trick of measuring some of the copies of the states before they go through the process to remove the issue of systematic errors. This trick alone is not really a new idea (it is present in GST and [15], for example, with a different algorithm that estimates the gates used for the preparation of the states rather than the states themselves), but it is possible only with a QPT algorithm that does not have to use a particular set of predetermined input states. While such QPT algorithms can be found in the literature ([1,6–8], for example), none of them have been tailored for unitary processes. Therefore, as mentioned above, they suffer from the drawbacks of increased complexity compared to our approach: more parameters to estimate, more input states, no pure states.

We focus on mitigating the impact of the systematic errors because, as [11] pointed out, the systematic errors are, in practice, the only errors (on the input states) if the gates that are used for the input state preparation are unitary. This is because the input states are prepared by initializing all qubits at $|0\rangle$, which yields a pure state (we assume that the assignment to $|0\rangle$ is at least repeatable), and the unitary gates applied to $|0\rangle \otimes \dots \otimes |0\rangle$ for the preparation of the input states (those “state-preparation gates” are to be distinguished from the unitary gate to be identified by QPT) preserve the purity of their input state. The pure states provided by the state-preparation unitary gates can have wrong values (this is the systematic error), but they do not change every time we repeat the experiment by definition (otherwise they would be mixed states). Assuming that the gates used for the preparation of the input states are unitary is reasonable, because if unitary gates cannot be realized, then unitary QPT is pointless. Reference [15] also

highlighted the importance of systematic errors for general QPT.

We intended to use a pure state QST algorithm from the literature for our QST-based QPT method, but there are surprisingly few articles about pure state QST. The work of Goyeneche *et al.* [23] is closest to fit our needs, but, unfortunately, it uses entangled measurements that cannot be performed by measuring each qubit individually. We want to use measurements that are as simple as possible because, as stressed in [15], we can never be sure that the model we have for the measurements is accurate. But the simpler, the measurement the more we can rely on the model. In [14] we introduced two original QST setups that use only unentangled measurements. In the present paper, we use one of them to perform QST for each measured state. We chose unentangled measurements because single-qubit gates are more reliable than multiqubit gates, and unentangled measurements can be realized with single-qubit quantum gates and a measurement in the computational basis. Like all QPT algorithms that do not rely on GST, we choose to consider that those measurements are realized without errors (or at least, that the errors in the measurements are small enough for the QPT estimate to be reasonably accurate). We assume that some kind of measurement calibration like in [12,24,25] (also known as quantum detector tomography or quantum measurement tomography) has been performed.

Our contributions in the present paper are as follows. Section II explains how to achieve unitary QPT using only the results of the QST of the measured states. This QPT could work with any QST algorithm, but we recommend using the measurements and QST algorithm described in Appendix A. In Sec. III we provide a necessary and sufficient condition on the measured states for unitary QPT to be possible, and we show that our QPT algorithm always works when this condition is satisfied (in the absence of QST error). In Sec. IV we present our recommendations on how to prepare the initial states to be sure that the above-mentioned condition is satisfied. In Sec. V we apply our algorithm to simulated data. Finally, in Sec. VI we assess the performance of our algorithm for data from two trapped-ion qubits.

II. PROPOSED QST-BASED QUANTUM PROCESS TOMOGRAPHY METHOD

A. Notations and data model

An n_{qb} -qubit pure state $|\varphi\rangle$ can be decomposed in the d -element computational basis $|0\dots 0\rangle, |0\dots 01\rangle, |0\dots 10\rangle, \dots, |1\dots 1\rangle$ ($d = 2^{n_{qb}}$). The components of $|\varphi\rangle$ in the basis can be stored in a d -element vector $\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix}$. The components v_j are complex and $\sum_{j=1}^d |v_j|^2 = 1$. The global phase of $|\varphi\rangle$ has no physical meaning.

In the rest of the paper we will use the vector notation instead of the kets (except for $|0\rangle$, which will still be used for one qubit). Contrary to most papers dealing with QPT, we will not use mixed states or density matrices as we are considering pure states in a closed system. We need the system to be closed, because all processes of a closed system are unitary.

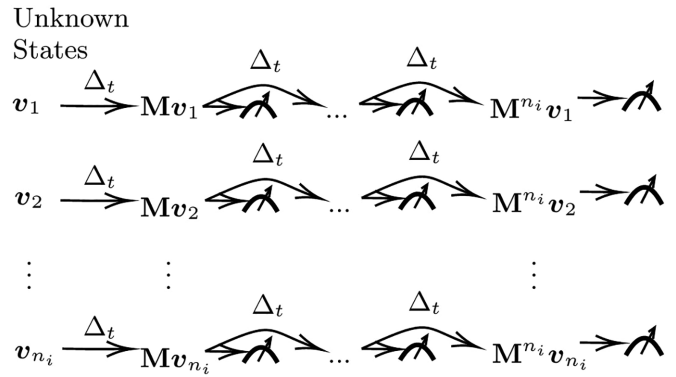


FIG. 1. QPT setup. The “double arrows” symbolize that some copies are measured (straight arrow) and the others are fed through the next gate.

The unitary matrix that characterizes the process to be identified is denoted as \mathbf{M} , and it is unique up to a global phase. Most often, the gate is physically realized by applying a constant Hamiltonian \mathbf{H} to the input state in a closed system. After a time delay Δ_t the state vector is multiplied by $\mathbf{M} = \exp \frac{-i}{\hbar} \mathbf{H} \Delta_t$ according to the Schrödinger equation (\hbar is the reduced Planck constant, i is the imaginary unit, \exp is the matrix exponential).

The QPT algorithm that we use relies on estimates of the measured states. Those estimates are computed with a QST algorithm. In Appendix A we summarize the QST algorithm that we recently introduced in [14] that relies on only one-qubit Pauli measurements performed in parallel on each qubit. A total of $2n_{qb} + 1$ types of those measurements need to be performed on each measured state.

B. QPT setup

Our QPT algorithm is designed to be robust to systematic errors on the initial states. To this end, we assume that the initial states are unknown but that measurements are performed at different time steps; see Fig. 1. The number of initial states is n_i . They are measured after waiting $\Delta_t, 2\Delta_t, \dots$ or $n_s \Delta_t$ (n_s is the maximum number of time delays), and the state vector is multiplied by the matrix \mathbf{M} associated with the process from one time delay to the next. The number of types of measurements that are performed on each qubit is called n_t . With the QST algorithm of Appendix A, we need $n_t = 2n_{qb} + 1$. One instance of a measurement alone does not bring much information on the state: We know only which one of the d possible outcomes has occurred. This is why we choose to perform each type of measurement n_c times in to estimate the probabilities of all outcomes. Each copy of the input states can be measured only once, therefore we need $n_s n_t n_c$ copies of each one of the n_i input states, for a total of $n_i n_s n_t n_c$ prepared input states.

The initial states (v_1, \dots, v_{n_i}) are never measured directly. We could imagine a similar setup where they are measured and the number of considered time delays is decreased by one (in fact, we used this approach in [22] with $n_s = 2$ and $n_i = d$). We chose not to use this approach here because some current quantum computers do not allow one to measure some states right after they are prepared. This is the case for the

computer we used in Sec. VI. This is explained in Sec. VI (more precisely in the description of Fig. 11) where we detail the implementation of our algorithm on a cloud quantum computing platform.

The results of the experiment are the measurement counts. For each one of the $n_i \times n_s$ measured states with each one of the n_r types of measurements, we count the number of times each one of the d outcomes occurred. Those d counts sum to n_c , and each one of those groups of d measurement counts can be modeled as a random variable that follows a multinomial distribution with n_c trials and the probabilities of the d outcomes are determined by the value of the associated measured state and the type of measurement. Table II in Appendix D shows an example of measurement counts for a given QPT setup with $n_i = 4$, $n_s = 2$, $n_r = 5$.

C. Main algorithm

We assume that QST is performed properly for the measured states of Fig. 1. The measured states are denoted with the following convention:

$$\mathbf{v}_{j,k} = \mathbf{M}^k \mathbf{v}_j, j \in \{1, \dots, n_i\}, k \in \{1, \dots, n_s\}, \quad (1)$$

and $\{\widehat{\mathbf{v}}_{j,k}\}_{j,k}$ are their estimates. Those QST estimates each have a phase indeterminacy $\theta_{j,k}^{QST}$ and a QST error $\mathbf{e}_{j,k}^{QST}$ such that $E(\|\mathbf{e}_{j,k}^{QST}\|_2) \xrightarrow{n_c \rightarrow +\infty} 0$ (E is the expected value):

$$\widehat{\mathbf{v}}_{j,k} = \mathbf{M}^k \mathbf{v}_j e^{i\theta_{j,k}^{QST}} + \mathbf{e}_{j,k}^{QST} \quad \begin{matrix} j \in \{1, \dots, n_i\} \\ k \in \{1, \dots, n_s\} \end{matrix} \quad (2)$$

For the rest of this paper, when we refer to the ‘‘QST error,’’ we mean $\mathbf{e}_{j,k}^{QST}$, not $\theta_{j,k}^{QST}$. For the rest of this section, we consider that there is no QST error: $\mathbf{e}_{j,k}^{QST} = 0$ unless stated otherwise. In Sec. II A we stated that the global phases of the states have no physical meaning since states that differ by only a global phase are the same. This is why we do not consider $\theta_{j,k}^{QST}$ to be an ‘‘error’’: it affects our QST estimate even in the ideal case with an infinite number of measurements and with no source of error.

We know that

$$\mathbf{v}_{j,k+1} = \mathbf{M} \mathbf{v}_{j,k} \quad j \in \{1, \dots, n_i\}, k \in \{1, \dots, n_s - 1\}. \quad (3)$$

For the sake of simplicity, we define

$$\begin{aligned} \mathbf{X} &= [\mathbf{v}_{1,1}, \dots, \mathbf{v}_{1,n_s-1}, \mathbf{v}_{2,1}, \dots, \mathbf{v}_{n_i,n_s-1}], \\ \mathbf{Y} &= \mathbf{M}\mathbf{X} = [\mathbf{v}_{1,2}, \dots, \mathbf{v}_{1,n_s}, \mathbf{v}_{2,1}, \dots, \mathbf{v}_{n_i,n_s}], \\ \widehat{\mathbf{X}} &= [\widehat{\mathbf{v}}_{1,1}, \dots, \widehat{\mathbf{v}}_{1,n_s-1}, \widehat{\mathbf{v}}_{2,1}, \dots, \widehat{\mathbf{v}}_{n_i,n_s-1}], \\ \widehat{\mathbf{Y}} &= [\widehat{\mathbf{v}}_{1,2}, \dots, \widehat{\mathbf{v}}_{1,n_s}, \widehat{\mathbf{v}}_{2,1}, \dots, \widehat{\mathbf{v}}_{n_i,n_s}]. \end{aligned} \quad (4)$$

With those notations, (3) becomes $\mathbf{Y} = \mathbf{M}\mathbf{X}$, and we ignore the setup of Fig. 1 and imagine that we are dealing with the simpler setup of Fig. 2 (with the convention that \mathbf{x}_ℓ is the ℓ th column of \mathbf{X}).

This representation is closer to the SQPT setup (known initial states, measured output). In fact, the only difference is that here the virtual input states (states on the left-hand side of Fig. 2) are not set to predetermined values, but are prepared with unknown quantum gates (including the gate that we are trying to identify). Those input states are estimated from the

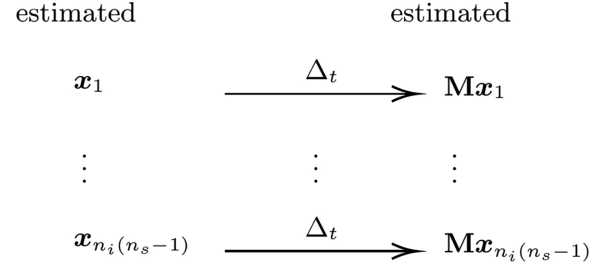


FIG. 2. Virtual QPT setup.

measurements. The algorithm that we will now describe could work with an SQPT setup with known predetermined input states \mathbf{X} and measured output states \mathbf{Y} . But not all SQPT algorithms would work with the ‘‘real’’ original setup (Fig. 1) because they often require the input states to be set to predetermined values.

When we state that the virtual input states (i.e., the states represented by the columns of \mathbf{X}) ‘‘are informationally complete’’ or ‘‘the process is identifiable,’’ we mean that there is only one unitary quantum process that transforms the virtual input states represented by the columns of \mathbf{X} into the virtual output states represented by the columns of \mathbf{Y} . If this is not the case, then there are several quantum processes which are compatible with the measured states in the columns of \mathbf{X} and \mathbf{Y} , i.e., there are several solutions to the QPT problem, and any QPT algorithm would fail to find a unique solution.

One could think that the problem of finding the unitary process that transforms \mathbf{X} into \mathbf{Y} is a simple linear least square problem with a unitarity constraint. But this is not the case because the columns of \mathbf{X} and \mathbf{Y} are known only up to global phases. Those global phases are not physical, but, when solving for \mathbf{M} up to a global phase, the difference between the phases of the columns of \mathbf{X} and those of \mathbf{Y} matter.

Let us define the relative phases between the \mathbf{x}_ℓ and the \mathbf{y}_ℓ :

$$\xi_\ell = \theta_{j,k}^{QST} - \theta_{j,k+1}^{QST}, \quad \begin{matrix} j \in \{1, \dots, n_i\}, k \in \{1, \dots, n_s - 1\} \\ \ell = k + (n_s - 1)(j - 1) \\ \ell \in \{1, \dots, n_i(n_s - 1)\} \end{matrix} .$$

We can rewrite (3) with $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$ (which are known from the QST):

$$e^{i\xi_\ell} \widehat{\mathbf{y}}_\ell = \mathbf{M} \widehat{\mathbf{x}}_\ell \quad \ell \in \{1, \dots, n_i(n_s - 1)\}, \quad (5)$$

where $\widehat{\mathbf{x}}_\ell$ and $\widehat{\mathbf{y}}_\ell$ are the ℓ th columns of $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$, respectively.

For any index ℓ_0 , changing \mathbf{M} to $\mathbf{M}e^{-i\xi_{\ell_0}}$ and ξ_ℓ to $\xi_\ell - \xi_{\ell_0} \forall \ell$ does not change the equality (5). Therefore, we can also assume that a given ℓ_0 (we explain how to choose it in Sec. II D) is such that $\xi_{\ell_0} = 0$ taking into account the fact that \mathbf{M} can be recovered up only to a global phase.

In the next section, we explain how to obtain the estimated phase factors $e^{i\xi_\ell}$. From that, we can define

$$\widetilde{\mathbf{y}}_\ell = \widehat{\mathbf{y}}_\ell e^{i\xi_\ell} \quad \ell \in \{1, \dots, n_i(n_s - 1)\}, \quad (6)$$

with which an estimate of \mathbf{M} can easily be found (if $n_i(n_s - 1) \geq d$) as the problem becomes

$$\widetilde{\mathbf{Y}} = \mathbf{M} \widehat{\mathbf{X}} \quad (7)$$

with

$$\widetilde{\mathbf{Y}} = [\widetilde{\mathbf{y}}_1, \dots, \widetilde{\mathbf{y}}_{n_i(n_s-1)}]. \quad (8)$$

In the absence of systematic error, $\widehat{\mathbf{M}} = \widetilde{\mathbf{Y}}\widehat{\mathbf{X}}^\dagger$ is a solution ($[\cdot]^\dagger$ is the pseudo-inverse). But it is generally not a unitary solution because of the QST errors. Finding $\widehat{\mathbf{M}}$ in the ensemble of d -dimensional unitary matrices [denoted as $\mathbb{U}_d(\mathbb{C})$] that is the total least square solution of (7) has been solved by Arun [26] (with the caveat that that work solves, with different notations, $\widetilde{\mathbf{Y}}^* = \widehat{\mathbf{X}}^*\mathbf{M}$, where $[\cdot]^*$ is the transconjugate, instead of $\widetilde{\mathbf{Y}} = \mathbf{M}\widehat{\mathbf{X}}$). Arun's solution can be rewritten as

$$\begin{aligned} \mathbf{B} &= \widetilde{\mathbf{Y}}\widehat{\mathbf{X}}^*, \\ \mathbf{U} \mathbf{S} \mathbf{V}^* &= \mathbf{B}, \\ \widehat{\mathbf{M}}_{LS} &= \mathbf{U} \mathbf{V}^*, \end{aligned} \quad (9)$$

where $\mathbf{U} \mathbf{S} \mathbf{V}^*$ is the singular value decomposition of \mathbf{B} . This is optimal in the total least square sense, meaning that $\widehat{\mathbf{M}}_{LS}(\widehat{\mathbf{X}} + \Delta_X^0) = (\widetilde{\mathbf{Y}} + \Delta_Y^0)$ where Δ_X, Δ_Y are the solutions to the following optimization problem:

$$\{\Delta_X^0, \Delta_Y^0\} = \arg \min_{\{\widehat{\mathbf{X}} + \Delta_X, \widetilde{\mathbf{Y}} + \Delta_Y\} \in \mathcal{L}} \|\Delta_X\|^2 + \|\Delta_Y\|^2, \quad (10)$$

where \mathcal{L} is the ensemble of the pairs of $d \times n_i(n_s - 1)$ matrices linked by a unitary transformation: $\mathcal{L} = \{\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{d \times n_i(n_s - 1)}, \exists \mathbf{P} \in \mathbb{U}_d(\mathbb{C}), \mathbf{Y} = \mathbf{P}\mathbf{X}\}$.

We use the total least square approach because both $\widehat{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ are subject to errors. Δ_X^0 and Δ_Y^0 can be understood as our estimates of those errors. This approach would be optimal in the maximum likelihood sense if the errors on $\widehat{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ were Gaussian iid on every component. This is not the case in practice (especially if $n_s > 2$: in this case, some columns of $\widehat{\mathbf{X}}$ are also in $\widetilde{\mathbf{Y}}$ up to a phase, thus their errors are highly correlated), but minimizing the norm of the error is a relevant approach as a first approximation.

The solution $\widehat{\mathbf{M}}_{LS}$ is unique if and only if both $\widehat{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ have full rank. This is not explicitly stated in [26], but it is proven for the orthogonal Procrustes problem in [27], which [26] showed to be equivalent to the total least squared problem that we consider.

D. Phase recovery algorithm

The aim of the current section is to find $\{e^{i\widehat{\xi}_\ell}\}_\ell$ given the vectors, $\widehat{\mathbf{x}}_\ell, \widehat{\mathbf{y}}_\ell$ $\ell \in \{1, \dots, n_i(n_s - 1)\}$ such that there exists a unitary matrix \mathbf{M} that realizes (5).

We will use the fact that unitary matrices preserve the dot product:

$$\widehat{\mathbf{x}}_{\ell_1}^* \widehat{\mathbf{x}}_{\ell_2} = (\mathbf{M}\widehat{\mathbf{x}}_{\ell_1})^* (\mathbf{M}\widehat{\mathbf{x}}_{\ell_2}) = \widehat{\mathbf{y}}_{\ell_1}^* \widehat{\mathbf{y}}_{\ell_2} e^{i(\widehat{\xi}_{\ell_2} - \widehat{\xi}_{\ell_1})}. \quad (11)$$

Therefore, for any $\{\ell_1, \ell_2\}$ pair in $\{1, \dots, n_i(n_s - 1)\}^2$, such that $\widehat{\mathbf{y}}_{\ell_1}^* \widehat{\mathbf{y}}_{\ell_2} \neq 0$, we have the following estimate of $\widehat{\xi}_{\ell_2} - \widehat{\xi}_{\ell_1}$:

$$\widehat{\xi}_{\ell_1, \ell_2} = \arg \left(\frac{\widehat{\mathbf{x}}_{\ell_1}^* \widehat{\mathbf{x}}_{\ell_2}}{\widehat{\mathbf{y}}_{\ell_1}^* \widehat{\mathbf{y}}_{\ell_2}} \right), \quad (12)$$

where \arg is the phase of a complex number. Using (12), we can compute estimates $\widehat{\xi}_{\ell_2}$ of all the phases $\xi_{\ell_2} \forall \ell_2 \in \{1, \dots, n_i(n_s - 1)\}$ relative to a single phase ξ_{ℓ_0} (by setting ℓ_1 to ℓ_0). Using the fact that $\xi_{\ell_0} = 0$ the estimate of $\widehat{\xi}_{\ell_2}$ is

$$\widehat{\xi}_{\ell_2} = \widehat{\xi}_{\ell_0, \ell_2}. \quad (13)$$

The problem of the choice of ℓ_0 remains. To solve it, let us look at (12) with ℓ_1 replaced by ℓ_0 (this is how we compute

$\widehat{\xi}_{\ell_0, \ell_2}$):

$$\widehat{\xi}_{\ell_0, \ell_2} = \arg \left(\frac{\widehat{\mathbf{x}}_{\ell_0}^* \widehat{\mathbf{x}}_{\ell_2}}{\widehat{\mathbf{y}}_{\ell_0}^* \widehat{\mathbf{y}}_{\ell_2}} \right). \quad (14)$$

It assumes that the dot products $\widehat{\mathbf{x}}_{\ell_0}^* \widehat{\mathbf{x}}_{\ell_2}$ and $\widehat{\mathbf{y}}_{\ell_0}^* \widehat{\mathbf{y}}_{\ell_2}$ are not zero. In practice, for $\widehat{\xi}_{\ell_0, \ell_2}$ to be a good estimate, we need both dot products to be as far from zero as possible. Interestingly, the two dot products are supposed to have the same modulus [see (11)]; therefore, we choose the index ℓ_0 solution of

$$\ell_0 = \arg \max_{\ell} \left(\min_{\ell_2} |\widehat{\mathbf{y}}_{\ell}^* \widehat{\mathbf{y}}_{\ell_2}| \right). \quad (15)$$

With this choice, ℓ_0 is such that the smallest dot product is as high as possible. The optimization is performed by exhaustive search.

In practice, if the corresponding maximum is 0 i.e., if, for all the $\widehat{\mathbf{y}}_{\ell_2}$, we can find an orthogonal $\widehat{\mathbf{y}}_{\ell_2}$, then this approach fails as (14) cannot be written for all the phases. In practice, two vectors are never going to be truly orthogonal, so in our implementation of our method, we consider that two unit-norm vectors are orthogonal when the modulus of their dot product is smaller than b_{orth} initially set to 0.05. And if the quantity maximized in (15) is greater than this b_{orth} , we simply compute all the phases with (14) and then (13). Otherwise we apply the following algorithm:

(1) We start by computing the phase differences $\widehat{\xi}_{\ell_0, \ell_2}$ (14) such that $\widehat{\mathbf{y}}_{\ell_2}$ are not orthogonal (modulus of dot product $> b_{orth}$) to $\widehat{\mathbf{y}}_{\ell_0}$. We then have the absolute phase of $\widehat{\mathbf{y}}_{\ell_2}$ using (13).

(2) We call \mathcal{F} the set of $\{\widehat{\mathbf{y}}_{\ell_2}\}_{\ell_2}$ for which we do not have the phase yet. We define \mathcal{S} as the complement of \mathcal{F} in $\{\widehat{\mathbf{y}}_{\ell_2}\}_{\ell_2}$.

(3) For each element $\widehat{\mathbf{y}}_{\ell_f}$ of \mathcal{F}

(a) If all elements of \mathcal{S} are orthogonal (modulus of dot product $< b_{orth}$) to $\widehat{\mathbf{y}}_{\ell_f}$ we change nothing and go to the next $\widehat{\mathbf{y}}_{\ell_f}$.

(b) Else, we define $\widehat{\mathbf{y}}_{\ell_s}$ as the element of \mathcal{S} that is the least orthogonal (greatest modulus of dot product) to $\widehat{\mathbf{y}}_{\ell_f}$.

(c) We compute the relative phase $\widehat{\xi}_{\ell_s, \ell_f}$ with (12) and deduce the 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) We remove $\widehat{\mathbf{y}}_{\ell_f}$ from \mathcal{F} and add it to \mathcal{S} .

(4) If \mathcal{F} is empty, then the algorithms successfully ends.

(5) If \mathcal{F} is not empty but the number of elements in it has decreased since Step 3, then we go to Step 3.

(6) If \mathcal{F} is not empty and the number of elements did not change, but \mathbb{C}^d is spanned by the elements of \mathcal{S} , then we remove the elements of \mathcal{F} from $\widehat{\mathbf{Y}}$ and the corresponding elements of $\widehat{\mathbf{X}}$, exit the phase recovery algorithm, and go on to solve (9) without the elements we removed.

(7) If the conditions in Step 5 and 6 are false, and b_{orth} is still 0.05, then we change b_{orth} to 0 and go to Step 3.

(8) If b_{orth} was already 0 when reaching Step 7, then this is a failure case and the algorithm ends.

This algorithm cannot stay in an infinite loop for the following reasons:

(1) The maximum number of times we can go from Step 5 to Step 3 without going to Step 6 is the cardinal of \mathcal{F} at Step 2 (because the cardinal of \mathcal{F} decreases by at least one every time), which is strictly smaller than $n_i(n_s - 1)$.

(2) The maximum number of times we can go from Step 7 to Step 3 is 1.

The algorithm can fail at Step 8, but we will show in Sec. III A that it always works if the states represented by the columns of \mathbf{X} are informationally complete.

E. Application to the standard QPT setup

The reader might be more familiar with the SQPT setup where the input states are known and only one time delay is considered. Our algorithm can easily be adapted to this setup because, as we stated in Sec. II C, the virtual setup of Fig. 2 is almost the SQPT setup, with $n_x = n_i(n_s - 1)$ input states: x_1, \dots, x_{n_x} . The only difference between the two is that, for the SQPT setup, the input states are considered to be known (and not estimated by QST). This is not a problem for our QPT algorithm: in the SQPT setup we can still define the matrices \widehat{X} (because the input states are known) and \widehat{Y} (because the output states are estimated by the QST algorithm), compute \widetilde{Y} with the algorithm of Sec. II D, and, finally, compute \widehat{M}_{LS} with (9).

III. IDENTIFIABILITY CONDITIONS

A. A necessary and sufficient condition for process identifiability

The QST error $\{e_{j,k}^{QST}\}_{j,k}$ is neglected in the current section, therefore $\widetilde{Y} = \mathbf{M}\widehat{X}$, with $\widetilde{Y} = \widehat{Y}\mathbf{D}(\xi)$, and $\mathbf{D}(\xi)$ is the diagonal matrix defined by the phases contained in ξ : $\mathbf{D}(\xi) = \begin{pmatrix} e^{i\xi_1} & & \\ & \ddots & \\ & & e^{i\xi_{n_x}} \end{pmatrix}$ [n_x is the number of columns in \widehat{X} , so $n_i(n_s - 1)$ in the base problem].

Therefore, the QPT problem after QST consists of finding the unitary matrix \mathbf{M} subject to the following condition:

$$\widehat{Y}\mathbf{D}(\xi) = \mathbf{M}\widehat{X}, \quad (16)$$

where \widehat{X} and \widehat{Y} are known from the QST and the elements ξ_1, \dots, ξ_{n_x} of ξ that define $\mathbf{D}(\xi)$ are unknown before the QPT. \widehat{X} contains our estimates of the virtual input states. Since there is no QST error, the columns of \widehat{X} are the same as those of \mathbf{X} up to global phases.

The following condition on \mathbf{X} is necessary and sufficient for the states represented by the columns of \mathbf{X} to be informationally complete with the setup of Fig. 1:

$$\forall \ell \in \{1, \dots, n_x\}, \text{rank}(\mathbf{F}_S^{n_x}(\mathbf{x}_\ell)) = d, \quad (17)$$

where \mathbf{F}_S is the function that takes as its input a matrix \mathbf{X}_{in} whose columns are columns of \mathbf{X} and that returns the columns of \mathbf{X} (grouped in a matrix in the order in which they appear in \mathbf{X}) that are not orthogonal to at least one column of \mathbf{X}_{in} . $\mathbf{F}_S^{n_x}$ is \mathbf{F}_S applied n_x times, and \mathbf{x}_ℓ is the ℓ th column of \mathbf{X} . This definition is closely linked to our phase recovery algorithm. If we change the starting b_{orth} to 0, then, at Step 2, the columns of \mathcal{S} match the columns of $\mathbf{F}_S^1(\mathbf{x}_{\ell_0})$. They match them in the sense that there are as many columns and their positions in \widehat{Y} and \mathbf{X} , respectively, are the same. This is because $\widehat{y}_{\ell_1} \perp \widehat{y}_{\ell_2} \Leftrightarrow \mathbf{x}_{\ell_1} \perp \mathbf{x}_{\ell_2}$ (\perp means that two vectors are orthogonal). The k th time we go to Step 5, the columns of \mathcal{S} match the columns of $\mathbf{F}_S^{k+1}(\mathbf{x}_{\ell_0})$, and, importantly, the subspaces they span have the same dimension.

Equation (17) is a condition on the columns of the matrix \mathbf{X} . However, it is very easy to check that we could have set this condition on the columns of \widehat{X} , \mathbf{Y} , or \widehat{Y} and have an equivalent condition. This is because multiplying all the columns by the same unitary matrix on the left or multiplying each of them by a different scalar phase factor does not change the rank or the orthogonality between the columns. We use \mathbf{X} because it is a matrix that contains the input states (of the virtual setup in Fig. 2), and we prefer to have a condition on the input states.

Appendix B shows that, in the absence of QST error, (17) is a necessary (Sec. B 2) and sufficient (Sec. B 1) condition for the states represented by the columns of \mathbf{X} to be informationally complete (i.e., for \mathbf{M} to be identifiable up to a global phase). Furthermore, the proof of the sufficiency of (17) in Sec. B 1 also shows that our QPT algorithm always succeeds if (17) is true.

This is a strong validation of our algorithm: It always achieves QPT if the setup (i.e., the measured states) makes it possible and fails only if the states represented by the columns of \mathbf{X} are informationally incomplete (i.e., if QPT is also impossible for any other algorithm). Of course, this is true only if there are no QST errors. When considering non-null QST errors, we can have issues with setups for which \mathbf{X} is poorly conditioned and for which \mathbf{X} contains groups of columns that are too close to being orthogonal for the phase recovery to succeed.

B. A simpler sufficient condition

We have established that (17) is a necessary and sufficient condition for the states represented by the columns of \mathbf{X} to be informationally complete. The condition of (17) is quite cumbersome to check however, and we prefer to use the following sufficient condition (it may be shown that it is not necessary):

$$\text{rank}(\mathbf{X}) = d \quad \text{and} \quad (18)$$

$$\exists \ell_0 \in \{1, \dots, n_x\}, \quad \forall \ell \in \{1, \dots, n_x\} \quad \mathbf{x}_{\ell_0} \not\perp \mathbf{x}_\ell.$$

In plain words, \mathbf{X} has full rank and there exists a column of \mathbf{X} that is not orthogonal to any of the others.

It is very easy to check that (18) \Rightarrow (17): if (18) is met, then all the columns of \mathbf{X} are in $\mathbf{F}_S^k(\mathbf{x}_\ell)$ for any $k \geq 2$ and any ℓ .

Therefore, when we design the QPT setup of Fig. 1 we have to hope (or to make sure) that the quantum states that are represented by the columns of \mathbf{X} satisfy (17) or (18).

In practice, (18) is a very reasonable condition. The probability of two random states (with any nondegenerated density function) being orthogonal is 0 and the probability of d random vectors or more (in a d -dimensional Hilbert space) being in a subspace of dimension $\leq d - 1$ (this relates to the condition on the rank of \mathbf{X}) is also 0. Essentially the states that make our algorithm fail are in a set of zero measure.

Therefore the conditions of (18) [and the condition of (17)] will always be satisfied in practice. Even if we try to prepare states that make our QPT method fail, the small random error in their preparation will ensure that the actual states satisfy (18). This does not mean that we can safely ignore the conditions though. Indeed, if \mathbf{X} has full rank but is almost singular, or if there are too many columns close to being orthogonal, then our algorithm (that works on \widehat{X} instead of

the unknown \mathbf{X} is expected to yield low performance if there are QST errors.

IV. QPT SETUPS COMPATIBLE WITH IDENTIFIABILITY

A. Our recommendations for suitable QPT setups

If possible, we recommend considering the assumed behavior of the gate that we are trying to identify (otherwise we hereafter propose a solution suited to any unitary gate without prior knowledge). For example, if the gate to be identified is

supposed to be a two-qubit CNOT gate, $\mathbf{M}_{tg} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

(“ tg ” stands for target), then, choosing a single input state ($n_i = 1$) and $d + 1 = 5$ time steps ($n_s = 5$) is a very bad option. This is because the CNOT gate applied twice is supposed to return the initial state. Therefore, \mathbf{X} has two identical pairs of columns. In practice, it will make $\widehat{\mathbf{X}}$ close to being singular and the quality of the estimate of \mathbf{M} will be very poor.

To have a matrix \mathbf{X} such that (18) is “comfortably” satisfied with the CNOT gate, let us consider $n_s = 2$ time delays, and $n_i = d = 4$ input states that form a basis of the Hilbert space with one of them far from being orthogonal to all the others. For example, we can aim for the following targets:

$$\begin{aligned} \mathbf{v}_1^{tg} &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & \mathbf{v}_2^{tg} &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix}, \\ \mathbf{v}_3^{tg} &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, & \mathbf{v}_4^{tg} &= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \end{aligned} \quad (19)$$

The actual states $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ should be reasonably close to those targets, but our QST algorithm will behave as if the states were totally unknown (so that we are robust to systematic errors).

Using the fact that multiplying two vectors by the same unitary matrix preserves their dot product, it is very easy to check that $\mathbf{X} = \mathbf{M}[\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3 \ \mathbf{v}_4]$ satisfies (18) and thus (17) with a comfortable margin for any unitary matrix \mathbf{M} if the $\{\mathbf{v}_k\}_k$ are equal (or fairly close) to their target.

The target states of (19) can be generalized to any number n_{qb} of qubits:

$$[\mathbf{v}_1^{tg}, \dots, \mathbf{v}_d^{tg}] = \underbrace{\begin{pmatrix} 1 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} 1 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix}}_{n_{qb} \text{ times}} \quad (20)$$

in the sense that the target vectors on the left-hand side are defined as the columns of the matrix on the right-hand side. For all n_{qb} -qubit unitary gates, using the states of (20) with $n_i = d, n_s = 2$ generates a matrix \mathbf{X} that satisfies (18) because its columns form a basis and none of them are orthogonal. These states have the advantages of being unentangled and very easy to prepare: If all qubits are initialized at $|0\rangle$, then the

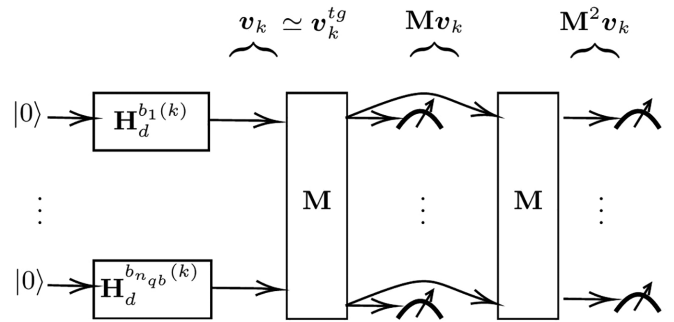


FIG. 3. Quantum circuit representing the preparation of a state \mathbf{v}_k^{tg} of (20) and the semiblind QPT of \mathbf{M} using this state. The circuit has to be realized for all $k \in \{1, \dots, d\}$. The 2×2 matrix \mathbf{H}_d is the unitary matrix associated with the one-qubit Hadamard gate, and $b_j(k) \in \{0, 1\}$ is the j th element of the binary decomposition of $k - 1$ over n_{qb} bits. Thus $\mathbf{H}_d^{b_j(k)} = \mathbf{I}_2$ if $b_j(k) = 0$ and $\mathbf{H}_d^{b_j(k)} = \mathbf{H}_d$ if $b_j(k) = 1$. This setup works well for QST for any value of the unitary matrix \mathbf{M} that represents the gate to be identified. Our algorithm is robust to poor implementations of the Hadamard gates. The “double arrows” in the middle symbolize that half of the copies are measured (straight arrow) and the other half fed through the next gate (in practice, this is achieved by waiting $2\Delta_t$ instead of Δ_t).

k th state can be prepared by applying a one-qubit Hadamard gate to the qubits with indices for which there is a 1 in the binary decomposition of $k - 1$. For example, for \mathbf{v}_1^{tg} , all qubits remain equal to $|0\rangle$ and no Hadamard gate is applied, for \mathbf{v}_2^{tg} , all qubits are initialized at $|0\rangle$ and a Hadamard gate is applied to the last qubit, etc. The Hadamard gate is represented by the unitary matrix $\mathbf{H}_d = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$; it transforms $|0\rangle$ into $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$.

The setup of Fig. 3 with $n_i = d, n_s = 2$ generates the initial states of (20). It is very interesting to note the following:

(1) As explained above, it can identify any type of unitary gate without prior knowledge [the sufficient condition (18) is always satisfied].

(2) It is fairly easy to prepare, as we require only a single type of gate (Hadamard) other than the gate that we want to identify.

(3) We can also tolerate errors in the Hadamard gates: they do not have to be perfect and they do not have to be identical, we simply require the behavior of each gate to remain unchanged while we prepare copies of each state.

(4) Using such simple gates at most once for each qubit should limit the decoherence issues.

One drawback of the setup of Fig. 3 is that there can be issues with the conditioning of \mathbf{X} when n_{qb} increases. The matrix \mathbf{X} is always invertible with the states of (20). But we observed that, experimentally, its smallest singular value decreases exponentially (with n_{qb}) towards 0. For more than four qubits we recommend considering more than d input states or using one-qubit rotation gates (instead of Hadamard gates) with adapted angles that provide a good conditioning and maintain the nonorthogonality condition in (18). The conditioning and the inner product of the columns of \mathbf{X} do not depend on the value of the unitary matrix \mathbf{M} , they depend on only the values of the initial states.

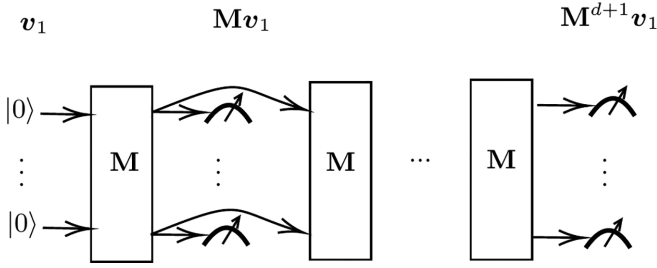


FIG. 4. Quantum circuit that can be used to perform QPT of the gate represented by \mathbf{M} with a single input state and $d + 1$ time steps: $n_i = 1$, $n_s = d + 1$. With this setup, the QPT is possible if and only if the matrix \mathbf{M} generates states $\mathbf{X} = [\mathbf{M}\mathbf{v}_1, \dots, \mathbf{M}^d\mathbf{v}_1]$ that satisfy (17). This setup is very easy to realize as it uses only one value of the initial state. The “double arrows” symbolize that the $n_c n_i$ copies are measured (straight arrow) and the others are fed through the next gate.

This setup can be less efficient than the one described below (Fig. 4) if we have a good idea of the behavior of the gate that we are trying to identify. For example, with $n_{qb} = 2$, if the unitary process that we want to identify is supposed to be represented by

$$\mathbf{M}_{tg} = \frac{1}{2} \begin{pmatrix} 1 & -\sqrt{2} & 0 & 1 \\ 1 & \sqrt{2} & 0 & 1 \\ 1 & 0 & -\sqrt{2} & -1 \\ 1 & 0 & \sqrt{2} & -1 \end{pmatrix}, \quad (21)$$

we hereafter show how to exploit this knowledge to design a more efficient setup. If \mathbf{M} is close enough to \mathbf{M}_{tg} , then we can use a single initial state, $n_i = 1$ and $n_s = 5$ time delays, and \mathbf{M} should (unless it is very far from the target) make the initial state evolve in a way that makes (18) true. If the initial state is $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, for example, then we can compute the matrix \mathbf{X} we would have if \mathbf{M} was exactly the target:

$$\mathbf{X}_{tg} = [\mathbf{M}_{tg}\mathbf{v}_1, \dots, \mathbf{M}_{tg}^d\mathbf{v}_1]. \quad (22)$$

The matrix \mathbf{X}_{tg} is well conditioned (the greatest singular value only 2.5 times greater than the smallest), and its first column is reasonably far from being orthogonal to the others (smallest modulus of dot product of 0.14). This setup (generalized to any number of qubits) is represented in Fig. 4. Considering fewer states and more time delays than in the setup of Fig. 3 means that we partly rely on the gate we are trying to identify to create the states that we will use. The drawbacks of this approach are the following:

(1) We are never sure that (17) is true with a comfortable margin, unless we have a good enough prior knowledge about the behavior of the gate that we want to characterize. In contrast, with $n_i = d$, $n_s = 2$ and the initial states of (20), we are sure that (18) and (17) are comfortably satisfied for any \mathbf{M} .

(2) With most of the “classic” (i.e., most often considered and used in the literature) quantum gates, the setup of Fig. 4 yields a matrix \mathbf{X} that does not satisfy (17). This is because those gates often involve 90° rotations (which can make too

many columns of \mathbf{X} close to being orthogonal) or do not change some directions of the Hilbert space (which can make \mathbf{X} poorly conditioned).

(3) If all the gates \mathbf{M} of Fig. 4 are not identical, the estimate will be degraded, whereas if the two gates to be identified in Fig. 3 are different, it is not really a problem, because the second gate is identified, whereas the first one is used only for the state preparation. We think that this is not a big problem because the way the gates are physically realized (see Sec. II A) makes it easy to apply the same gate several times.

(4) Having higher values of n_s can create decoherence issues for some architectures because the state is observed after waiting $n_s \Delta_t$.

But the setup of Fig. 4 has the following advantages:

(1) Preparing copies of a single input state value is much simpler for the operator. Only one type of gate (the gate to be identified) is used. Using more types of gates is problematic because, even though the QPT algorithm of Sec. II makes no assumption on the values of the unitary matrices that represent each gate (\mathbf{M} and the gates used for the initial states preparation), it still assumes that they are unitary gates that operate in the exact same way every time we repeat the experiment. If this is not the case, the quality of our estimate of \mathbf{M} will suffer.

(2) Having a higher n_s means that more of the estimated output states are being “reused” as estimates of input states and vice versa. This means that fewer states are measured overall, and that we can afford to make more copies of each state that we measure. For example, with $n_{qb} = 2$, we can use the $n_i = 4$ different initial states values of (20) and $n_s = 2$ time steps to estimate a two-qubit gate. This requires eight different states to be measured. If the same gate can be estimated with a single input state value ($n_i = 1$) and $n_s = 5$ time delays, this requires only five different states to be measured.

(3) The example of the target gate (21) that we used to illustrate a case where $n_i = 1$, $n_s = d + 1$ might seem far fetched and make the reader think that we had to find a particular matrix \mathbf{M} to make the setup of Fig. 4 ($n_i = 1$, $n_s = d + 1$) work. This is not the case; the setup of Fig. 4 yields good performance for random unitary gates (see Sec. V D).

We could consider an intermediate setup with $d > n_i > 1$. It can be useful if \mathbf{M} is the identity matrix (up to a global phase) in a subspace of the Hilbert space but brings enough diversity to the supplement of this subspace. For example, let us assume that we want to perform QPT for a gate that

$$\text{is supposed to be represented by } \mathbf{M}_{tg} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}.$$

If we consider a single arbitrary input state value \mathbf{v}_1 and if $\mathbf{M} = \mathbf{M}_{tg}$, then the matrix \mathbf{X}_{tg} of (22) will never have full rank because its first two rows will contain the same value in all columns, so that these rows will be colinear. But if we consider $n_i = 2$ input states and $n_s = 3$ time delays, we can create informationally complete states

$$\text{with } \mathbf{v}_1^g = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{v}_2^g = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \text{ It is very easy to check that } \mathbf{X} =$$

$[\mathbf{M}\mathbf{v}_1, \mathbf{M}^2\mathbf{v}_1, \mathbf{M}\mathbf{v}_2, \mathbf{M}^2\mathbf{v}_2]$ satisfies (17) comfortably if \mathbf{M} and the input states are equal to their targets. There-

fore (17) should still be satisfied if they are close to their targets.

B. Comparison with the literature about unitary multiqubit QPT

Reich *et al.* in their Appendix 1 [10] give a necessary and sufficient condition on the input states for a unitary process to be uniquely (up to a global phase) determined among all processes (unitary or not). This condition is

$$\text{Com}(\{\rho_\ell\}_\ell) = \{e^{i\theta} \mathbf{I}_d\}_{\theta \in \mathbb{R}}, \quad (23)$$

where ρ_ℓ is the density matrix of the ℓ th mixed input state, Com is the comutant, $\text{Com}(\{\rho_\ell\}_\ell)$ refers to the ensemble of the unitary matrices that commute with all $\{\rho_\ell\}_\ell$, and \mathbf{I}_d is the $d \times d$ identity matrix.

If we write (23) for pure input states, $\rho_\ell = \mathbf{x}_\ell \mathbf{x}_\ell^*$, the condition becomes

$$\text{Com}(\{\mathbf{x}_\ell \mathbf{x}_\ell^*\}_\ell) = \{e^{i\theta} \mathbf{I}_d\}_{\theta \in \mathbb{R}}. \quad (24)$$

The conditions (17) and (24) are equivalent, as shown in Appendix B 3. This does not mean that (17) is equivalent to the original condition (23) of Reich *et al.* because the latter was defined with mixed input states, and (24) is its reformulation with pure input states. Therefore we have the equivalence between (17) and (23) only when the input states are pure.

In Ref. [10] the authors give two examples of sets of input states that satisfy (23). The first one is a set of $d + 1$ pure input states: d states that form an orthonormal basis and a last state that is the average of the first d states (it works with any orthonormal basis). The same idea is present in (18): one of the states is not orthogonal to the others.

Reich *et al.* [10] also show that if we allow the use of mixed states, then there exists a set of only two informationally complete states. This is because, if a mixed state with d distinct eigenvalues goes through the unitary process, then the output has the same eigenvalues but the eigenvectors are multiplied by the unitary matrix associated with the process. Therefore it is possible to evaluate the unitary matrix on the orthonormal basis of the eigenvectors by using a single mixed input state. This is not enough though because all the states are orthogonal to one another, so they need a second input state. We chose not to consider mixed states because they are more complex to create and set to a predetermined value. Baldwin *et al.* [9] remark when they use the results of [10]: “In practice we do not have reliable procedures to produce a desired, reproducible, mixed state.” We do not fully agree with that statement, since any mixed state can be seen as a statistical mixture of at most d pure states (represented by the eigenvectors of the density matrix). If we can generate all the pure states of the mixture and then randomly select one of them for each copy of the mixed state that we generate (with the probability of choosing each eigenvector given by the associated eigenvalue of the density matrix of the target mixed state), then we can generate the desired mixed state. Generating a mixed state in this way would not be optimal, however, as it involves preparing copies of d different pure states and then “mixing” them (not recording which pure state is used for which copy of the mixed state). Using the d pure states directly is more efficient than using this “mixture.”

Beyond reformulating (24) as (17) and providing a set of simple input states, our contribution is that, contrary to [10], we do provide a QPT algorithm that works with every set of input states that verifies our necessary and sufficient condition. We also took advantage of the fact that (17) is loose to introduce our semiblind setup that removes the issue of systematic errors.

Baldwin *et al.* [9] propose a unitary QPT algorithm [after their Eq. (20)] adapted to the following specific set of d input states:

$$\left\{ \delta_1, \left\{ \frac{1}{\sqrt{2}}(\delta_1 + \delta_k) \right\}_{k \in \{2, \dots, d\}} \right\} \quad (25)$$

(where δ_k is the d -dimensional vector that contains $d - 1$ zeros and a single 1 on the k th element). The last $d - 1$ states of the set are entangled; they can be more challenging to prepare precisely than unentangled states. This algorithm is much simpler than ours, but it works only for this particular set of d states, and the quality of the QPT’s estimate will be limited by the systematic errors on the initial states. Our algorithm works for any informationally complete set of state, and we take advantage of this property to propose a semiblind setup that eliminates the issue of systematic errors and works with the unentangled initial states of (20). Baldwin *et al.* also go further and propose an algorithm adapted to processes close to being unitary. This algorithm could actually be used with arbitrary input states, as long as they satisfy (17). But it is a l_1 minimization algorithm with a regularization parameter, and the solution (and its rank) will depend on the value of the regularization parameter.

V. PERFORMANCE ON SIMULATED DATA

We chose to first focus on the setup with the four initial states (19) to estimate random two-qubit gates to test the robustness of our method to different kinds of errors. We consider this setup because two-qubit QPT is a classic problem and the four initial states (19) are versatile. We aim to test the robustness of our method to the multinomial error (error generated by the finite number of measurements) in Sec. V A. Then, in Secs. V B and V C, we test its robustness to the systematic errors on the input states, and we compare our algorithm to that of Baldwin *et al.* [9]. Finally, in Sec. V D we expand the tested setup by increasing the number of qubits. We also compare the performance of the setup of Fig. 3 to that of the setup of Fig. 4 when the number of qubits is varied.

Other simulations were also performed. They introduced decoherence which is not taken into account by our model of measurement (the measured states are assumed to be pure). Those simulations are deemed less important and are provided in Appendix C.

A. Impact of the number of measurements

We designed simulations to determine the number n_c (the number of copies of each state that need to be prepared and measured for each measurement type) that is required to obtain an accurate estimate of \mathbf{M} with the setup of Fig. 3. We start by considering the case when the finite number of measurements is the only source of errors. There is no deco-

herence, the measurements follow the model, and the initial states that we consider are prepared with perfect Hadamard gates (as will be seen in Sec. VB, the last point does not really matter). We vary n_c from 20 to 25 000, and for each value of n_c we generate 5000 random quantum gates defined by unitary matrices created by applying the Gram-Schmidt algorithm to a random complex matrix with each coefficient i.i.d. and following the circularly symmetric centered complex normal distribution. For each n_c we also generate 5000 (one per gate to be identified) sets of $n_s n_t n_r$ random measurement counts (the measurement counts are defined at the end of Sec. IIB) associated with all n_c copies. They are simulated with a d -outcome multinomial distribution with the probabilities of all outcomes set to their theoretical values (Born rule). For example, if one of the measured states is $\mathbf{v} = (0.5 \ 0.5 \ 0.5i \ 0.5i)^T$ (to be identified with the measurements) and we perform measurements in the computational basis (measurement type ZZ in Appendix A) with $n_c = 50$ copies of this state measured in this basis, we simulate measurement counts by sampling a multinomial distribution with probability parameters given by the Born rule $\mathbf{p} = |\mathbf{I}_4 \mathbf{v}|^2 = (0.25 \ 0.25 \ 0.25 \ 0.25)^T$ for the four outcomes and $n_c = 50$ trials. The resulting empirical measurement counts could be $(13 \ 10 \ 15 \ 12)^T$, for example.

For each n_c , we therefore have 5000 unitary matrices $\{\mathbf{M}\}$ to be estimated and 5000 sets of measurement counts to perform the QPT. We run our algorithm and get 5000 estimates $\{\widehat{\mathbf{M}}_{LS}\}$. We then have to quantify the error between the matrices $\{\mathbf{M}\}$ and the matrices $\{\widehat{\mathbf{M}}_{LS}\}$. We choose to use the following metric denoted $\epsilon(\widehat{\mathbf{M}}_{LS}, \mathbf{M})$ and called the error:

$$\epsilon(\widehat{\mathbf{M}}_{LS}, \mathbf{M}) = \frac{1}{\sqrt{2d}} \|\mathbf{M} - \widehat{\mathbf{M}}_{LS} e^{i\phi}\|, \quad (26)$$

where ϕ is the angle that minimizes the error (it accounts for the fact that \mathbf{M} can be recovered only up to a global phase) and $\|\cdot\|$ is the Frobenius norm. The values of this metric range from 0 (if $\widehat{\mathbf{M}}$ and \mathbf{M}_{true} are equals up to a global phase) to 1 (if they are orthogonal with respect to the Hilbert-Schmidt inner product). It can be shown that

$$\phi = \arg(\text{tr}(\widehat{\mathbf{M}}_{LS}^* \mathbf{M})) \quad (27)$$

(arg is the phase of a complex number and tr is the trace).

In the literature, the fidelity $f(\widehat{\mathbf{M}}_{LS}, \mathbf{M})$ is more often used [see, e.g., Eq. (24) in [9]]. It is defined for all quantum processes (not only unitary processes). It can be shown that the two metrics are linked when we are dealing with unitary processes: $f(\widehat{\mathbf{M}}_{LS}, \mathbf{M}) = 1 - \epsilon(\widehat{\mathbf{M}}_{LS}, \mathbf{M})^2$. We consider that ϵ is much more attractive than f because ϵ is a distance and has a real meaning (a value ϵ of 0.1 can be seen as a 10% error). It is also much more informative when the estimate $\widehat{\mathbf{M}}_{LS}$ starts to become close to \mathbf{M} : clearly, an error ϵ of 0.1 is significantly worse than an error of 0.01, but comparing the associated values of f , equal to 0.99 and 0.9999, is harder.

Figure 5 shows the box plots (first and last quartiles, median, 5th and 95th percentiles and outliers) of the 5000 samples of the error for each value of n_c . We also display the line proportional to $\frac{1}{\sqrt{n_c}}$ that fits some of the measured medians (see caption).

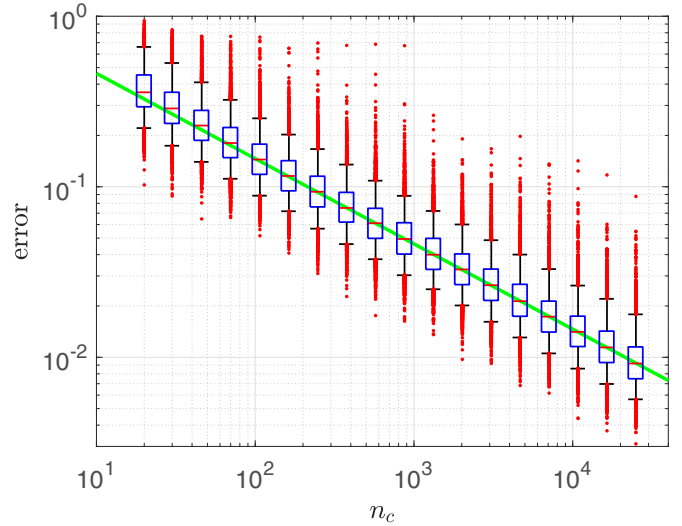


FIG. 5. Box plots of the QPT error with respect to the number n_c of measured state copies per state and per type of measurement is varied. The light (green) line from top left to bottom right represents the function $n \rightarrow \frac{C}{\sqrt{n}}$ where C is computed so that the line fits (in the least square sense) the median in the box plots associated with $n_c \geq 1000$. The central (blue) rectangle represents the interval ranging from the first and last quartiles where half of the observations are. The narrow (red) line in the middle of each rectangle is the median, and the (black) whiskers range from the 5th percentile to the 95th percentile. By definition, 90% of the samples are within the range of whiskers, whereas the ones that are not are called “outliers” and are represented as small (red) dots above and below each box plot.

With the log scale, we find the classic inverse relation between the estimation error and the square root of the number of samples (the line accurately fits the last medians) if n_c is high enough ($n_c \geq 100$).

B. Impact of the systematic errors

We design a second simulation to investigate the influence of the systematic errors present in the initial states. We simulate the initial states of (19), we fix $n_c = 1000$, and we simulate systematic errors so that each of the $n_t n_c n_s$ copies of a given state has the same error. Physically this means the initialization of the states at $|0\rangle$ or the Hadamard gate used to transform them has errors, but, for a given gate or initialization, this error is the same for each copy. We have to consider the multinomial error ($n_c = 1000$) in addition to the systematic error because the QPT error would always be zero otherwise. The systematic errors are modeled as random complex vectors with each coefficient i.i.d. following the circularly symmetric centered complex normal distribution added to each vector of (19). After adding this error we renormalize the initial vectors. The standard deviation of this systematic error is the same for all components, and we vary it from 0 to 0.3. We also test our algorithm with totally random states uniformly sampled on the ensemble of pure states. Figure 6 displays the box plots of the QPT error versus the standard deviation (std.) of the error. The infinite std. (inf) refers to the totally random initial states.

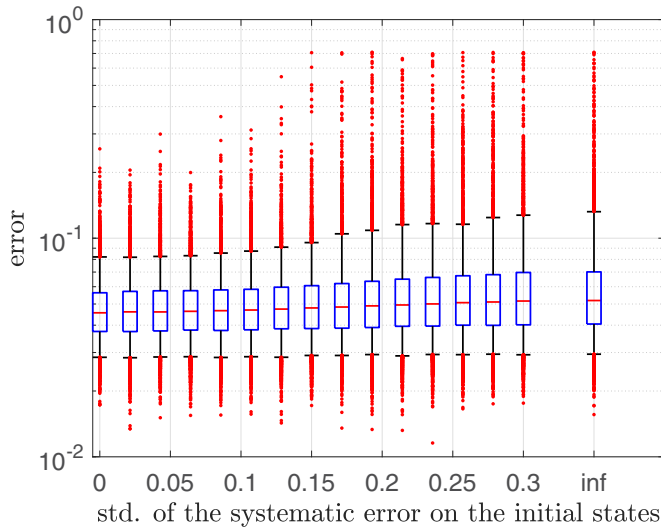


FIG. 6. Box plots of the QPT error vs the standard deviation (std.) of the systematic error on the initial states with $n_c = 1000$. The same box plot representation as in Fig. 5.

There is a significant trend of greater 95th percentile and greater outliers when the systematic error increases. The 95th percentile goes from 0.083 to 0.13. For the lower part of the box plots, the difference is less noticeable. Even when the initial states are totally random, the median of the error (0.052) is close to the median we obtained with no systematic error (0.046). The QPT algorithm is resilient to the systematic errors on the initial states because it makes no assumption on the values of the initial states. The only drawback of the systematic errors is that they can make the QPT problem harder by making too many initial states close to being orthogonal or making the matrix \mathbf{X} poorly conditioned. The chance of those problems occurring is fairly low, even with totally random input states. This is the reason why the impact of the systematic errors is mostly visible in the upper part of the box plots.

C. Comparison with the literature about the SQPT setup

In the current section, we aim to compare our algorithm with that of [9] in the case involving two qubits and four initial states ($n_{qb} = 2$, $n_i = 4$). It is not a trivial task as the algorithm of [9] has been designed for the SQPT setup of Fig. 7 (in the case $n_{qb} = 2$, $n_i = 4$). Our algorithm was defined for the semiblind setup of Fig. 8. We can adapt it to the SQPT setup of Fig. 7 (see Sec. II E), but we would lose our resiliency to systematic errors, which was one of our goals. We chose to compare the following three algorithms:

- (1) Our algorithm running on the setup of Fig. 8 with $n_c = 1000$.
- (2) The algorithm of [9] running on the setup of Fig. 7. We chose to set $n_c = 2000$ in order to account for the fact that Fig. 7 requires four states to be measured instead of eight.
- (3) Our algorithm running on the setup of Fig. 7 with the adaptations of Sec. II E and with $n_c = 2000$.

For those three options, the initial states are the states defined in Eq. (20) of [9] or in (25) in the current paper.

trusted

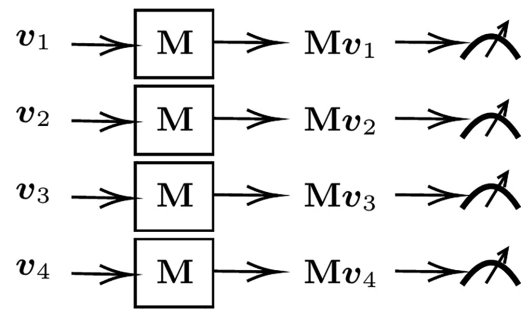


FIG. 7. SQPT setup with $n_{qb} = 2$.

We introduce a systematic error modeled in the same way as the systematic error in the previous section. For the sake of simplicity we use the QST algorithm of Appendix A for all QPT algorithms.

Let us first compare the left-hand plot with the other two in Fig. 9. When the systematic error increases, the error becomes smaller in the left-hand plot. This is unsurprising, since our algorithm for the semiblind setup (whose performance is represented on the left-hand plot) has been designed to be resilient to the systematic errors; it does not use the values of the initial state. In contrast, the other two algorithms assume that the values of the initial states are exactly known. It also makes sense that for lower systematic errors, the nonblind algorithms running on the SQPT setup (they are represented on the middle and right-hand side of Fig. 9) perform better because they use the values of the initial states rather than wasting half the measurements to get a noisy estimate of two sets of states before and after the gate has been applied. The middle plot of Fig. 9 (algorithm of [9] applied to the SQPT setup of Fig. 7) contains higher errors than the right-hand plot for all values of the std. of the systematic error (except for the “infinite std.,” where they are identical). This means that our algorithm adapted to the SQPT setup yields a better estimate than that of [9]. This is because the latter is very simple and elegant, but it was not designed to mitigate the effect of the errors that we model here (systematic and multinomial error). It directly estimates the coefficients of the unitary matrix from the QST estimates of the states. The resulting matrix has no reason to be unitary if there are errors. In contrast, our algorithm finds the unitary matrix that fits our QST estimate best (in the least square sense). The algorithm of [9] is simpler

unknown

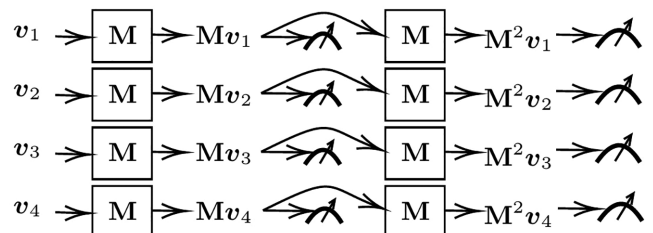


FIG. 8. Semiblind setup with $n_{qb} = 2$, $n_i = 4$, $n_s = 2$.

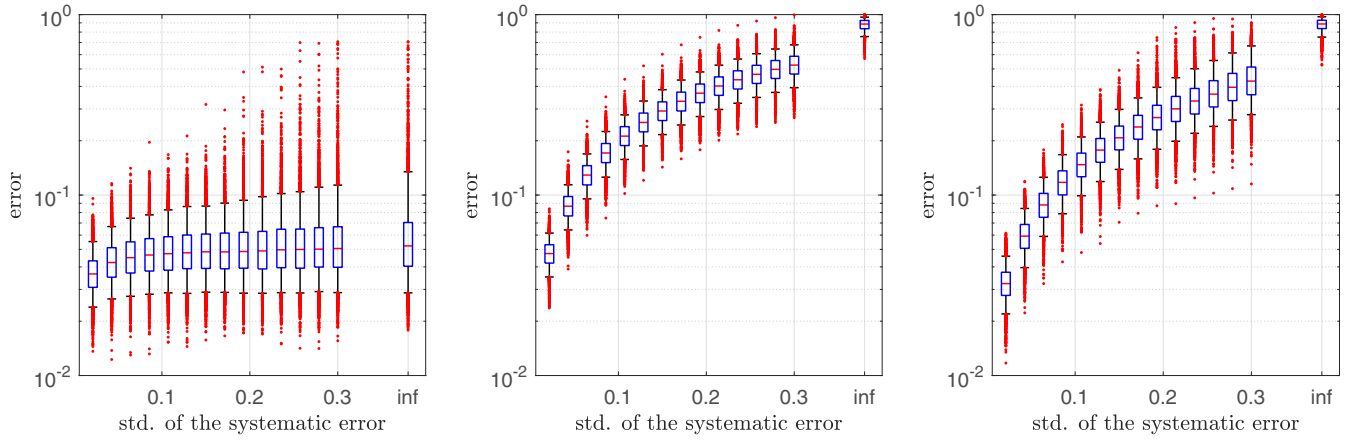


FIG. 9. Box plots of the QPT error in the presence of systematic errors with the initial states of [9]. The first plot (left-hand side) represents the error with our semibland algorithm running on the semibland setup of Fig. 8. The second plot (middle) represents the error with the algorithm proposed in [9] applied to the standard setup considered in [9] (represented in Fig. 7 for two qubits). The third plot (right-hand side) represents the performance of our algorithm adapted to run on the SQPT setup of Fig. 7.

and faster, but as we will see in Sec. V D, our QPT algorithm is very quick, at least compared to the QST algorithm we use (it is so quick, in fact, that its execution time is negligible compared to the QST).

Overall, the performances of our algorithm for the semibland setup are very satisfying. It yields lower errors than the algorithm of [9] (resp. than our algorithm with the adaptations of Sec. II E) when the standard deviation of the Gaussian systematic error is roughly 0.007 (resp. 0.025) on each component of the initial states. Those values (0.007 and 0.025) are obtained by linearly interpolating the median of the box plots in the three graphs and computing the values of systematic errors for which the interpolated lines of the median error cross.

D. Other setups

In this section we aim to investigate how well our QPT algorithm works with more than two qubits and with the setups of Fig. 3 and Fig. 4.

For one to six qubits, we simulate 500 random $d \times d$ unitary matrices by applying the Gram-Schmidt process to random Gaussian matrices (like in the previous subsections). And we try to identify the processes associated with each matrix with the following two setups:

(1) With the d initial states of (20) and two time delays ($n_i = d$, $n_s = 2$, displayed in Fig. 3). We simulate $n_c = 2500$ measurements per measurement type and per measured state. We consider that the states are prepared with the setup of Fig. 3. The Hadamard gates used for the initial state preparation are considered to be imperfect and are represented by $\begin{pmatrix} \cos(\theta_r) & -\sin(\theta_r)e^{i\phi_r} \\ \sin(\theta_r) & \cos(\theta_r)e^{i\phi_r} \end{pmatrix} \mathbf{H}_d$ instead of \mathbf{H}_d , where θ_r and ϕ_r are two random i.i.d. Gaussian centered angles with a standard deviation of 0.05 radians (two different values are sampled for each gate of Fig. 3 and for each one of the 500 gates to be identified).

(2) With a single random initial state and $d + 1$ time delays ($n_i = 1$, $n_s = d + 1$, like the setup of Fig. 4 but with a

random initial state). We use $n_c = \lceil 2500 \frac{2d}{d+1} \rceil$ ($\lceil \cdot \rceil$ refers to the closest integer), so that the total number of measurements $n_c n_i n_s n_t$ is the same (or almost the same if $2500 \frac{2d}{d+1}$ is not an integer) in both setups for a given number of qubits.

As explained in Sec. IV A, the first setup is more robust and the second can be easier to prepare and allows one to use higher n_c (since $\frac{2d}{d+1} > 1$).

The chosen n_c is very high for small values of n_{qb} . If $n_{qb} = 2$, for example, there are only $d = 4$ outcomes whose probabilities are to be estimated. Then having $n_c = 2500$ or $n_c = \lceil 2500 \frac{2 \times 4}{4+1} \rceil = 4000$ is more than enough to have very good estimates. But if $n_{qb} = 6$, there are $d = 64$ outcomes, and the associated probabilities are much smaller (since they

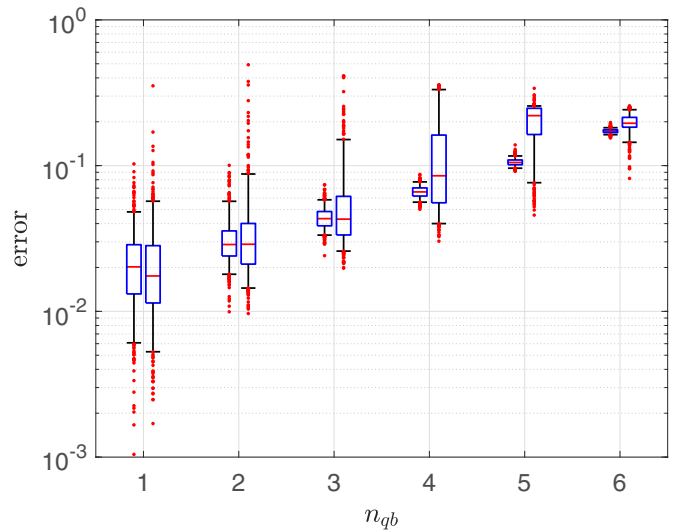


FIG. 10. Box plots of the QPT error for quantum gates acting on one to six qubits. There are two box plots for each number of qubits. The one on the left represents the errors with the first setup ($n_i = d$, $n_s = 2$), and the one on the right represents the errors with the second setup ($n_i = 1$, $n_s = d + 1$). Both setups have the same total number of measurements.

TABLE I. Average execution time of the QST of all measured states and of the QPT of the process for the setup of Fig. 3 (setup 1) and the setup of Fig. 4 (setup 2).

Algorithm \ n_{qb}	1	2	3	4	5	6
QST setup 1	0.06 s	0.14 s	0.37 s	1.67 s	12 s	188 s
QPT setup 1	7×10^{-5} s	9×10^{-5} s	1×10^{-4} s	3×10^{-4} s	6×10^{-4} s	2×10^{-3} s
QST setup 2	0.05 s	0.10 s	0.21 s	0.85 s	6.9 s	123 s
QPT setup 2	1×10^{-4} s	1×10^{-4} s	2×10^{-4} s	3×10^{-4} s	8×10^{-4} s	3×10^{-3} s

sum to one), then the number of measurements does not seem that excessive.

Figure 10 shows the box plots of the error for both setups and for all numbers of qubits. Unsurprisingly the error increases with n_{qb} . This is the case despite the fact that the total number of measurements $n_c n_i n_s n_t$ also increases with n_{qb} (n_c is constant but n_i increases for the first setup, for which $n_i = d$; n_s increases for the second setup, for which $n_s = d + 1$; and $n_t = 2n_{qb} + 1$ increases for both setups). This is not surprising because the number of parameters of the unitary matrix we estimate is $d^2 = 2^{2n_{qb}}$; it increases exponentially with n_{qb} .

For the first setup, the range of the error decreases (in log scale) when the number of qubits increases. We think this is because, when the number of parameters to be estimated (in \mathbf{M}) increases, the error on the matrix becomes more predictable as the errors on the parameters get “averaged” by the norm in (26). We do not observe this phenomenon with the second setup because for this setup we rely on \mathbf{M} to create a matrix \mathbf{X} that satisfies (17) with a good margin, and the fact that we choose 500 different random values of \mathbf{M} for each number of qubits makes the error vary much more. In contrast, the matrix \mathbf{X} of the second setup always satisfies (17) and does not depend on \mathbf{M} .

The relative performances of the two setups are interesting. For one to four qubits, with the second setup, the largest errors are larger and the smallest errors are smaller than with the first setup. This is in line with what we expected: The first setup is supposed to be more reliable, so it makes sense that its greatest outliers are smaller. The second setup uses the measurements more efficiently (see the end of Sec. IV A). It makes sense that, when we sampled a random \mathbf{M} that makes the second setup work, it works better than the first setup. And if we have an idea of the value of \mathbf{M} before performing the QPT, we will know in which part of the box plot the error is likely to be. Hopefully, for our \mathbf{M} , (17) will be satisfied with a comfortable margin, and we will know that the error is in the lower part of the box plot. The more qubits there are, the worse the second setup gets compared to the first. We think this is because for a random matrix (and \mathbf{X} is basically a random matrix for the second setup but is deterministic for the first setup if the systematic errors are neglected) (17) becomes closer to being false when dimension increases.

The average execution times of the QST and QPT algorithms are reported in Table I. The simulations were coded with Matlab and executed on a 210 Intel Xeon silver 4214 2.4 GHz. We allowed the script to run on 10 threads, but each simulation ran sequentially. Matlab only parallelized the linear algebra computation on large matrices.

Clearly, the execution time of the QPT is not significant, it is the QST that takes the longest. There are faster QST algorithms in the literature, and we could shorten the QST time greatly by not implementing the fine-tuning ML-based approach (see [14]). But we choose to sacrifice execution time for precision. In contrast, for the QPT, we need only to compute a few dot products for the phase recovery, and then a few products of $d \times d$ matrices, and then perform a singular value decomposition to solve the total least square problem under a unitarity constraint.

VI. RESULTS FOR A CNOT GATE OF A TRAPPED-ION QUANTUM COMPUTER

In this section we test our QPT algorithm using a trapped-ion quantum computer on Amazon Web Services (AWS) [28]. The test was performed in May 2022. AWS calls the device we used “Harmony” and advertises single-qubit gate fidelity of 0.9935, or $\epsilon = 0.0806$ with our definition of the error (26), a gate fidelity of 0.9602 (or $\epsilon = 0.1995$) for two-qubit gates,

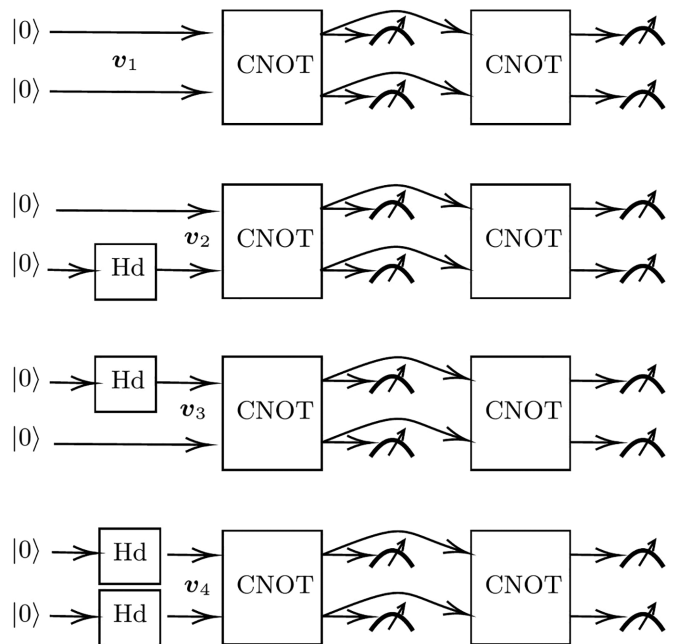


FIG. 11. QPT setup for a two-qubit CNOT gate, with $n_i = 4$ initial states and $n_s = 2$ time delays. In Sec. II B we stated that we apply the process to be identified before we measure any state, and that it is a choice that we had to make in order to implement the setup physically. Indeed, Ref. [28] does not let us measure the actual qubits that have supposedly been set to $|0\rangle$ and have never been modified by a quantum gate.

a coherence time of 1.667 s, and a state preparation and measurement (SPAM) characterized by a SPAM fidelity of 0.9961.

We want to perform QPT on a two-qubit CNOT gate. As stated in Sec. IV A, the use of two time steps is adapted to this gate. Therefore we chose $n_i = d = 4$, $n_s = 2$ with the four initial states $v_1^{i_s}$ to $v_4^{i_s}$ of (19), which are the same as (20) and Fig. 3, with $n_{qb} = 2$. As explained in Sec. IV A, those initial states can be created with Hadamard gates (see Fig. 3). In order not to rely on the implementation of the Hadamard gates, our algorithm will behave as if the input states were totally unknown. The target value of the process matrix of the CNOT gate to be identified is

$$\mathbf{M}_{rg} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (28)$$

This value is not used by our QPT method either. The process to be identified is considered to be totally unknown. The four circuits of Fig. 11 are implemented. Each of them is an explicit version of Fig. 3 (with $k = 1$ to $k = 4$) for $n_{qb} = 2$. The $n_t = 2n_{qb} + 1 = 5$ measurement types are performed $n_c = 250$ times on each one of the eight measured states. In total $n_c n_t n_s n_i = 250 \times 5 \times 2 \times 4 = 10\,000$ quantum mea-

surements are performed. Table II (in Appendix D) records the measurement counts for each outcome and each state.

We want to use the QST algorithm described in Appendix A to get an estimate of the measured states. We are facing an issue: the measurement types defined in Appendix A are ZZ, ZX, ZY, XX, and YX for two qubits, but the measurements that we have performed on the real quantum computer are ZZ, ZX, ZY, XX, and YY (the last measurement type is not the same). It turns out that the QST algorithm of Appendix A also works with those measurements. The initial estimate of the measured state will be wrong (because it works only with the former set of measurements), but the QST algorithm has two steps (see Sec. A 2), and the second step (fine tuning with maximum likelihood) corrects the initial estimate in the two-qubit case (because it can work with the types of measurements that are available). This is not a big problem for two reasons: (1) as just stated, the QST algorithm still works and (2) we aim to test the QPT algorithm, not the QST; the former relies on the QST output and does not use the measurements directly.

From the measurements, we estimate the eight measured states with the QST algorithm. We arrange the estimated states in the matrices $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ defined in Sec. II (they are estimates of $\mathbf{X} = [\mathbf{M}v_1, \mathbf{M}v_2, \mathbf{M}v_3, \mathbf{M}v_4]$ and $\mathbf{Y} = [\mathbf{M}^2v_1, \mathbf{M}^2v_2, \mathbf{M}^2v_3, \mathbf{M}^2v_4]$, respectively). Their numerical values are the following:

$$\hat{\mathbf{X}} = \begin{pmatrix} 1.00 - 0.00i & 0.76 - 0.00i & 0.70 - 0.00i & 0.49 - 0.00i \\ 0.01 - 0.01i & 0.65 - 0.01i & -0.06 + 0.05i & 0.45 - 0.00i \\ -0.02 - 0.03i & 0.02 - 0.03i & 0.06 - 0.03i & 0.54 - 0.06i \\ 0.03 - 0.06i & -0.01 - 0.08i & 0.68 - 0.19i & 0.50 - 0.02i \end{pmatrix},$$

$$\hat{\mathbf{Y}} = \begin{pmatrix} 1.00 - 0.00i & 0.72 - 0.00i & 0.70 - 0.00i & 0.54 - 0.00i \\ 0.01 + 0.01i & 0.70 + 0.06i & -0.02 - 0.00i & 0.46 + 0.04i \\ 0.03 - 0.00i & -0.02 - 0.01i & 0.72 - 0.02i & 0.51 + 0.08i \\ 0.00 - 0.01i & 0.01 - 0.01i & 0.01 - 0.02i & 0.47 + 0.08i \end{pmatrix}.$$

We then use the phase recovery algorithm of Sec. II D to compute $\tilde{\mathbf{Y}}$, which is the same as $\hat{\mathbf{Y}}$ but with each column multiplied by a phase factor. We then use the method of Sec. II C to find the unitary matrix $\hat{\mathbf{M}}_{LS}$ that links $\hat{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ (the rephased version of $\hat{\mathbf{Y}}$) best. Finally, we change its global phase by a factor $e^{i\phi}$ to compare it with \mathbf{M}_{rg} :

$\hat{\mathbf{M}}_{LS} \leftarrow e^{i\theta} \hat{\mathbf{M}}_{LS}$ with $\theta = \arg(\text{tr}(\hat{\mathbf{M}}_{LS}^* \mathbf{M}_{rg}))$, like in (27). This last step is possible (and useful) only if we know \mathbf{M}_{rg} , which is the case here. If we had no idea of what the gate was supposed to do, we would not perform this step.

The resulting (rounded) estimate is the following:

$$\hat{\mathbf{M}}_{LS} = \begin{pmatrix} 0.98 - 0.17i & -0.02 - 0.02i & 0.02 + 0.02i & 0.01 + 0.07i \\ 0.02 - 0.02i & 0.99 - 0.09i & 0.01 + 0.03i & 0.03 + 0.01i \\ 0.00 + 0.07i & -0.02 + 0.01i & 0.08 - 0.02i & 0.99 + 0.08i \\ -0.01 + 0.02i & -0.01 + 0.03i & 0.98 + 0.18i & -0.07 - 0.04i \end{pmatrix}.$$

The moduli are close to their target values, but there are fairly significant errors on the phases that cannot be corrected with a global phase shift. This is particularly noticeable between the first and third columns. The distance between $\hat{\mathbf{M}}_{LS}$ and the target \mathbf{M}_{rg} can be defined as $\epsilon(\hat{\mathbf{M}}_{LS}, \mathbf{M}_{rg}) \simeq 0.11$ with ϵ defined in (26). According to Fig. 5, 0.11 is a very reasonable error with $n_c = 250$ in this setup, so we cannot reject the hypothesis that the gate is perfectly realized by simply looking at the error. This is not our objective, however; we do

not want to perform quantum gate benchmarking but quantum gate tomography.

VII. CONCLUSION AND FUTURE WORK

In this paper we introduced a quantum process tomography (QPT) method that is very flexible with respect to the values of the initial states used (as long as they remain pure). We proposed a semiblind setup (Fig. 1) that splits the copies of

each initial state into n_s groups and measures the copies of the k th group (only once) after they go through the process $k \in \{1, \dots, n_s\}$ times. This trick allows us to estimate the states that go through the process rather than assuming that the input states are correctly prepared. The resulting QPT setup is resilient to systematic errors on the initial states. We proved that, in the absence of QST errors, our QPT algorithm always gives a perfect estimate of the process to be identified if we satisfy the condition of (17). We also showed that, if (17) is not satisfied, then it is not possible to find a QPT algorithm that works (because there are several different processes that yield the same measurement outcomes).

As explained in Sec. IIE our algorithm also applies to the more standard QPT (SQPT) setup (the values of the initial states are known beforehand, and all copies of the initial states are processed in the same way: they go through the process only once and then get measured). Our algorithm is attractive (since it is applicable to the semiblind and the SQPT setups) when we compare it to [9], which is the best-known multi-qubit unitary quantum process tomography algorithm (see the simulations in Sec. VC). We also tested the resilience of our algorithm to different types of errors, and we tested it to identify a CNOT gate of a trapped-ions qubit quantum computer.

Our QPT algorithm does not require an initial estimate of the process and is quite fast after the QST of the measured states has been performed; we have to compute only a few dot products (see Sec. IID) and to perform a singular value decomposition (see Sec. IIC). The limitation of our estimate is that it minimizes the least square error on the QST results (after the phase recovery), and as stated at the end of Sec. IIC, this is not optimal from a maximum likelihood standpoint. The least-square estimate of the process can be fine tuned with a slower but more precise gradient descent algorithm that starts at our estimates and finds the unitary matrix that maximizes the likelihood of the measurements without performing QST. Defining a model of the likelihood of the quantum measurements that takes into account the multinomial error as well as the potential nonpurity of the measured states is a challenge in and of itself, and we intend to study it in the future.

A proponent of gate set tomography (GST) could also criticize our QPT setup by pointing out the fact that it works only if the types of measurements performed on the unknown states are precisely known. This is a fair point, and the fact that the measurements that we use might be flawed is a drawback of our algorithm (it is also a drawback of all QPT algorithms in the literature that do not use GST) that we tolerated because we need a frame of reference. We trust neither the values of the input states nor the process to be identified, but we have to trust the measurement process, otherwise we would be left with the gauge error that plagues GST. This flaw in our algorithm is somewhat mitigated by the fact that we rely on only very simple unentangled measurements. We intend to mitigate it further by introducing a (blind) quantum detector tomography algorithm that can be used to estimate some parameters of the measurements that we propose to use for each qubit without using predetermined states. This is possible because the measurements that we perform are somewhat redundant.

APPENDIX A: STATES, MEASUREMENTS, AND QST

This Appendix quickly goes over the choices that we made for the measurements performed and the QST algorithm. We need it because the central QPT algorithm that we aim to present in this paper relies on some kind of QST. However, the choices we make here do not affect the QPT algorithm of Sec. II, and other types of measurements and QST can be used.

1. Definition of the measurements

We chose to perform measurements that have d outcomes (d -outcome measurements), as d is the maximum number of outcomes for a type of quantum measurement in a d -dimensional Hilbert space. We are not interested in the actual values of the outcomes: they can be denoted as $1, \dots, d$ or $0 \dots 0, 0 \dots 01, \dots, 1, \dots, 1$, it makes no difference to us. We are interested in the probabilities of the measurement outcomes. They are estimated by performing the measurements several times on copies of the considered state, and computing the frequencies of occurrence of each outcome.

If a projective measurement \mathcal{M} has d outcomes, then there exists a unitary matrix $\mathbf{E}_{\mathcal{M}}$ such that the probabilities of all outcomes when measuring any state \mathbf{v} are contained in the vector $|\mathbf{E}_{\mathcal{M}}^* \mathbf{v}|^2$ (with the convention that $|\cdot|^2$ is the element-wise squared modulus); this is known as the Born rule. The columns of the matrix $\mathbf{E}_{\mathcal{M}}$ (each of them is defined up to a global phase) entirely characterize the type of measurement (up to the values of the outcomes). We call it the eigenvector matrix, because its columns are the eigenvectors of the Hermitian matrix that is often used to characterize the measurements in the literature [see (2.102) in [3]].

We also choose to use unentangled measurements. Unentangled measurements are quantum measurements that can be performed in parallel on each qubit. To our knowledge unentangled measurements are the only types of measurements that can be performed on the current version of quantum computers without using entangling gates (that have to be characterized with QPT).

If the measurement \mathcal{M} is unentangled, then $\mathbf{E}_{\mathcal{M}}$ can be written as the tensor product of n_{qb} matrices in $\mathbb{U}_2(\mathbb{C})$. Those 2×2 matrices are the eigenvector matrices associated with each one-qubit measurement.

For each qubit, the 2×2 unitary matrices we use are among the following three:

$$\begin{aligned} \mathbf{E}_X &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, & \mathbf{E}_Y &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}, \\ \mathbf{E}_Z &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned} \quad (\text{A1})$$

If the qubit represents the spin of an electron, those eigenvector matrices represent the measurement of the spin component along directions X , Y , and Z .

For two or more qubits, measurements can be performed along direction X , Y , or Z for each qubit. It can be shown that the resulting eigenvector matrix is the tensor product of the two-dimensional matrices of (A1). For example, for

two qubits, performing a measurement along Z for the first qubit and along X for the second one yields the following eigenvector matrix:

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

In this example, if the qubits represent the spins of two electrons, then the measurement we perform is equivalent to measuring the first spin component along Z and the second along X . The spin pair measurement has four possible outcomes $(+\frac{1}{2}, +\frac{1}{2})$, $(+\frac{1}{2}, -\frac{1}{2})$, $(-\frac{1}{2}, +\frac{1}{2})$, and $(-\frac{1}{2}, -\frac{1}{2})$ in normalized units, and if \mathbf{v} represents the considered state, the probabilities of each outcome are in the vector $|\mathbf{E}_{ZX}^* \mathbf{v}|^2$.

There are $3^{n_{qb}}$ types of measurements with this definition. We will use only those characterized by the $2n_{qb} + 1$ eigenvector matrices contained in the following set:

$$\mathcal{E} = \left\{ \mathbf{E}_{\underbrace{Z \dots Z}_{n_{qb} \text{ times}}}, \left\{ \mathbf{E}_{\underbrace{Z \dots Z}_{n_{qb}-1 \text{ times}} [1] S \underbrace{X \dots X}_{i-1 \text{ times}}}, 1 \leq i \leq n_{qb} \right\}, S \in \{X, Y\} \right\}.$$

For example, with $n_{qb} = 2$, we have \mathbf{E}_{ZZ} , \mathbf{E}_{ZX} , \mathbf{E}_{ZY} , \mathbf{E}_{XX} , and \mathbf{E}_{YX} .

2. Quantum state tomography

In [14] we showed how a quantum state can be estimated by performing measurements for n_c copies of that state, successively with each measurement matrix of \mathcal{E} . The total number of measurements performed on copies of the state is $n_t n_c = (2n_{qb} + 1)n_c$. This might seem unimpressive as, for example, [23] would use only $4n_c$ or $5n_c$ copies (depending on the used basis). But the four or five measurement types of [23] are not all unentangled, and we were unable to find fewer than $(2n_{qb} + 1)$ types of unentangled measurements that make QST possible with a closed form explicit algorithm.

We here use the algorithm of Sec. 3 of [14] fined tuned with the maximum likelihood method of Sec. 4 with the Gaussian version of the likelihood. We use the Gaussian version instead of the real likelihood or the mixed algorithm because we want an algorithm that will “catch” types of errors that are not considered by the multinomial likelihood model (typically decoherence).

APPENDIX B: MATHEMATICAL DEMONSTRATIONS FOR THE IDENTIFIABILITY CONDITION

Our identifiability condition is (17). It is necessary and sufficient for \mathbf{M} to be identifiable (up to a global phase and with a given QPT setup) in the set of unitary matrices. In Sec. B 1 we show that it is a sufficient condition. Importantly we do this by showing that our algorithm is then able to retrieve \mathbf{M} up to a global phase. This means that our QPT algorithm works in any situation where QPT is theoretically possible. In Sec. B 2 we show that it is a necessary condition. Finally, in Sec. B 3 we show that (17) is equivalent to another condition of the literature: (24). In this Appendix we assume that there is no QST error (the columns of $\widehat{\mathbf{X}}$ and of $\widehat{\mathbf{Y}}$ are

the same as those of \mathbf{X} and of \mathbf{Y} , respectively, up to global phases).

1. Proof that (17) is sufficient

To show that (17) is sufficient for the states represented by the columns of \mathbf{X} to be informationally complete (or for “ \mathbf{M} to be identifiable”) with the setup of Fig. 1, we need to define what we mean by “the states represented by the columns of \mathbf{X} are informationally complete” (or “ \mathbf{M} is identifiable”). Formally, this means that \mathbf{X} [defined by \mathbf{M} and the \mathbf{v}_i ; see (4) and (2)] is such that the ensemble $\mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}})$ of unitary matrices that are compatible with the QST results $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$,

$$\mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}) = \{\mathbf{U} \in \mathbf{U}_d, \exists \xi \in \mathbb{R}^{n_x}, \mathbf{U} \widehat{\mathbf{X}} \mathbf{D}(\xi)^* = \widehat{\mathbf{Y}}\}, \quad (\text{B1})$$

contains only \mathbf{M} and matrices that are the same as \mathbf{M} up to a global phase: $\mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}) = \{e^{i\phi} \mathbf{M}\}_\phi$.

Expression (17) is a sufficient condition for the states represented by the columns of \mathbf{X} to be informationally complete because if (17) is true, then, essentially, the algorithms of Secs. II C and II D yield a unique \mathbf{M} up to a global phase:

(1) The algorithm of Sec. II D always succeeds, i.e., we exit the algorithm at Step 4 or 6. Indeed, if $\text{rank}(\mathbf{F}_S^{n_x}(x_{\ell_0})) = d$, then, even if the algorithm starts poorly and we have to set $b_{orth} = 0$ at Step 7, the condition of Step 6 will eventually be satisfied after going through Step 5 at most n_x times. This is because of the equality we pointed out in Sec. III A below (17) between the rank of $\mathbf{F}_S^k(x_{\ell_0})$ and the dimension of the subspace spanned by the columns of \mathcal{S} after going through Step 5 k times with $b_{orth} = 0$. The reader could think that we could exit the algorithm prematurely (i.e., before the set \mathcal{S} spans \mathbb{C}^d) because the condition that makes us loop from Step 5 to Step 3 stops being satisfied. This is a nonissue because if the condition is not satisfied, then it is pointless to continue as the elements of \mathcal{S} will not increase even if we were to go to Step 3.

(2) The phase differences between the columns of $\widetilde{\mathbf{Y}}$ that we exit the algorithm of Sec. II D with are the unique solution of (11) because (11) has a unique solution if and only if $\widehat{\mathbf{y}}_{\ell_1} \not\propto \widehat{\mathbf{y}}_{\ell_2}$ for the ℓ_1, ℓ_2 that we use to compute (12) at Step 3(c) of Sec. II D. We made sure that this is the case in the phase recovery algorithm.

(3) The ranks of the matrices $\widetilde{\mathbf{Y}}$ and $\widehat{\mathbf{X}}$ that we exit the algorithm of Sec. II D with are full (d), because (1) $\mathbf{F}_S^{n_x}(x_{\ell_0})$ only contains columns of \mathbf{X} by definition, thus (17) $\Rightarrow \text{rank}(\mathbf{X}) = d$; (2) \mathbf{M} is unitary, thus $\text{rank}(\mathbf{Y}) = \text{rank}(\mathbf{M}\mathbf{X}) = d$; and (3) there is no systematic error, thus $\text{rank}(\widehat{\mathbf{X}}) = \text{rank}(\mathbf{X}) = d$ and $\text{rank}(\widetilde{\mathbf{Y}}) = \text{rank}(\mathbf{Y}) = d$. Therefore, the total least square problem with a unitarity constraint has a single solution: \mathbf{M}_{LS} (as discussed at the end of Sec. II C). But any matrix that is the same as \mathbf{M}_{LS} up to a global phase is also a solution of the QPT problem. This is because an arbitrary choice has been made for the global phase of $\widetilde{\mathbf{Y}}$ in Sec. II D [ℓ_0 has been chosen with (15) and ξ_{ℓ_0} has been set to 0].

Therefore $\mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}) = \{e^{i\phi} \mathbf{M}\}_\phi$.

2. Proof that (17) is necessary

Let us now show that (17) is a necessary condition. We assume that \mathbf{X} is such that (17) is false, i.e.,

$$\exists \ell \in \{1, \dots, n_x\}, M(\mathbf{F}_S^{n_x}(\mathbf{x}_\ell)) < d. \quad (\text{B2})$$

We want to show that there exists a matrix \mathbf{M}_2 that differs from \mathbf{M} by more than a global phase such that $\mathbf{M}_2 \in \mathcal{U}(\mathbf{X}, \mathbf{MX})$ with the same definition for \mathcal{U} as in (B1). It is straightforward to show that, since there is no QST error, $\mathcal{U}(\mathbf{X}, \mathbf{MX}) = \mathcal{U}(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}})$. This means that the processes represented \mathbf{M}_2 and \mathbf{M} are different quantum processes (they differ by more than a global phase), but our measurements are not sufficient to distinguish them.

Let us first define the function \mathbf{G}_S and list its useful properties:

$$\mathbf{G}_S : k \longrightarrow \mathbf{F}_S^k(\mathbf{x}_\ell):$$

(1) The columns of $\mathbf{G}_S(k)$ are all also columns of $\mathbf{G}_S(k+1)$.

(2) If k_0 is the smallest integer such that $\mathbf{G}_S(k_0) = \mathbf{G}_S(k_0+1)$, then $\forall k \geq k_0, \mathbf{G}_S(k_0) = \mathbf{G}_S(k)$.

(3) The number of columns of $\mathbf{G}_S(k)$ is upper bounded by n_x (the number of columns of \mathbf{X}) $\forall k$.

Therefore the number of columns of $\mathbf{G}_S(k)$ increases strictly with k for the first k_0 iterations, and for $k \geq k_0$ \mathbf{G}_S becomes constant. And k_0 has to be smaller than n_x because the number of columns increases by at least one at each iteration before k_0 and is bounded by n_x .

For any ℓ , and in particular for the ℓ of (B2), $\mathbf{F}_S^{n_x}(\mathbf{x}_\ell) = \mathbf{F}_S^{n_x+1}(\mathbf{x}_\ell)$. Therefore, we can split the columns of \mathbf{X} into two groups \mathbf{X}_s [defined as $\mathbf{F}_S^{n_x}(\mathbf{x}_\ell)$] and \mathbf{X}_f (defined as the matrix that contains the other columns of \mathbf{X} in order). The matrix \mathbf{X}_f can be empty, but if it is not, the columns it contains are all orthogonal to the columns of \mathbf{X}_s [since $\mathbf{F}_S(\mathbf{X}_s) = \mathbf{X}_s$]. According to (B2), \mathbf{X}_s is not of full rank.

From there we consider the only two possible implications of (17) being false:

(1) \mathbf{X}_f is empty, $\mathbf{X}_s = \mathbf{X}$ and $\text{rank}(\mathbf{X}) < d$.

(2) \mathbf{X}_f is not empty and \mathbf{X} can be decomposed: $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_f] \mathbf{Q}_{per}$, where \mathbf{Q}_{per} is an $n_x \times n_x$ permutation matrix.

Either one of these conditions makes \mathbf{M} unidentifiable for the following reasons:

(1) If the condition of Case 1 is true, there exists a unit norm vector called \mathbf{v}_{ker} in the null space of \mathbf{X}^* , i.e., $\mathbf{X}^* \mathbf{v}_{ker} = \mathbf{0}$ and $\mathbf{v}_{ker}^* \mathbf{X} = \mathbf{0}$. We call \mathbf{V}_{hker} a $d \times (d-1)$ matrix such that $\mathbf{P}_v = [\mathbf{V}_{hker}, \mathbf{v}_{ker}]$ is a unitary matrix (\mathbf{V}_{hker} and \mathbf{P}_v are not unique for a given \mathbf{v}_{ker} ; \mathbf{V}_{hker} can be any orthogonal basis of the subspace orthogonal to \mathbf{v}_{ker}). We define $[\mathbf{C}_1 \ \mathbf{c}_2] = \mathbf{M} \mathbf{P}_v$ ($\mathbf{C}_1 \in \mathbb{C}^{d \times (d-1)}, \mathbf{c}_2 \in \mathbb{C}^d$), and a straightforward calculation shows that for any angle $0 < \phi < 2\pi$, the unitary matrix $\mathbf{M}_2 = [\mathbf{C}_1 \ \mathbf{c}_2 e^{i\phi}] \mathbf{P}_v^*$ is not the same as \mathbf{M} (even up to a global phase since $d > 1$) and \mathbf{M} and \mathbf{M}_2 are both in $\mathcal{U}(\mathbf{X}, \mathbf{MX})$ since $\mathbf{MX} = \mathbf{M}_2 \mathbf{X}$.

(2) If the condition of Case 2 is true, we have to consider only the case when \mathbf{X} is of full rank (because otherwise, we use the reasoning above), and $\text{vect}(\mathbf{X}_f)$ is the orthogonal complement of $\text{vect}(\mathbf{X}_s)$ (vect is the subspace spanned by the column vectors of a matrix). We define \mathbf{P}_s as an orthonormal basis of the subspace $\text{vect}(\mathbf{X}_s)$ and \mathbf{P}_f as an orthonormal basis of $\text{vect}(\mathbf{X}_f)$. We define the matrix \mathbf{M}_2 as $\mathbf{M}_2 = \mathbf{M}(\mathbf{P}_s \mathbf{P}_s^* e^{i\phi} +$

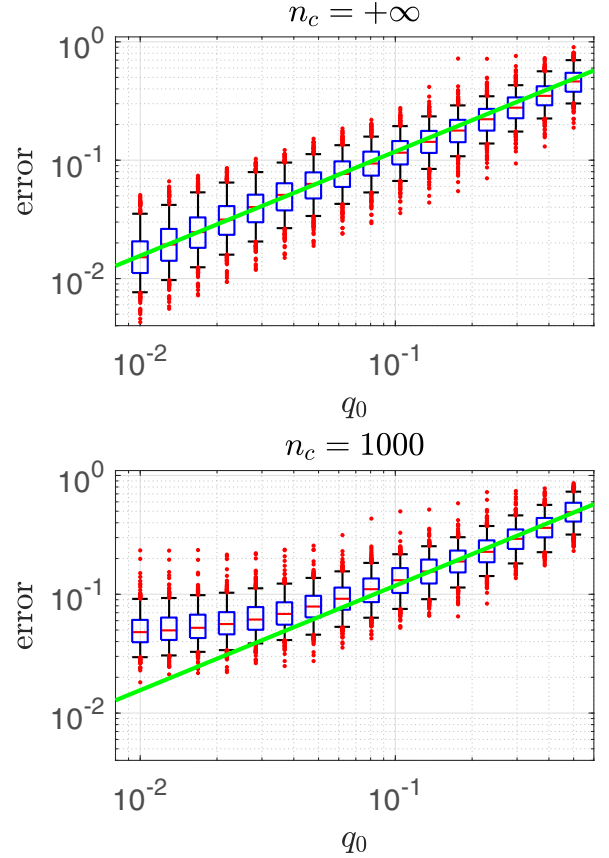


FIG. 12. Box plot of the QPT error with mixed input states. The setup of Fig. 3 is used with $n_c = 1000$ and with no multinomial error ($n_c = +\infty$). In the top plot, the (green) line from the bottom left to the top right is computed with a linear regression to fit the medians associated with $q_u < 0.1$. The same line is then duplicated in the lower plot. Both the slope and the intercept of the line are computed with the linear regression. This differs from the line of Fig. 5, where the slope was set to 0.5.

$\mathbf{P}_f \mathbf{P}_f^*$), and define $\mathbf{X}_{alt} = [\mathbf{X}_s e^{-i\phi} \ \mathbf{X}_f] \mathbf{Q}_{per}$ (\mathbf{Q}_{per} defined above). With those definitions \mathbf{M}_2 is unitary (this is easy to check), and a straightforward calculation shows that $\mathbf{MX} = \mathbf{M}_2 \mathbf{X}_{alt}$, \mathbf{X}_{alt} has the same columns as \mathbf{X} up to global phases. Let $\boldsymbol{\xi}$ be the vector that contains those phases, and $\mathbf{D}(\boldsymbol{\xi})$ be the diagonal matrix such that $\mathbf{X}_{alt} = \mathbf{X} \mathbf{D}(\boldsymbol{\xi})$. We have $\mathbf{MX} = \mathbf{M}_2 \mathbf{X}_{alt} = \mathbf{M}_2 \mathbf{X} \mathbf{D}(\boldsymbol{\xi})$; thus \mathbf{M} and \mathbf{M}_2 are both in $\mathcal{U}(\mathbf{X}, \mathbf{MX})$ but \mathbf{M}_2 and \mathbf{M} differ by more than a global phase.

3. Equivalence between (17) and (24)

The condition of Reich *et al.* has been rewritten in (24); it guarantees the identifiability of the process represented by \mathbf{M} among all processes. This is stronger than identifiability among unitary processes, which we guarantee with (17). Therefore (24) implies (17). Let us show that (17) implies (24):

Let $\mathbf{C} \in \mathcal{U}_d(\mathbb{C})$ be a unitary matrix in $\text{Com}(\{\mathbf{x}_\ell \mathbf{x}_\ell^*\}_\ell)$. Let us show that, if (17) is met, then $\exists \theta, \mathbf{C} = e^{i\theta} \mathbf{I}_d$.

If two matrices commute, then there exists a basis that diagonalizes both of them (see Theorem 1.3.12 in [29]). Therefore, the fact that \mathbf{C} commutes with all $\{\mathbf{x}_\ell \mathbf{x}_\ell^*\}_\ell$ implies that all $\{\mathbf{x}_\ell\}_\ell$

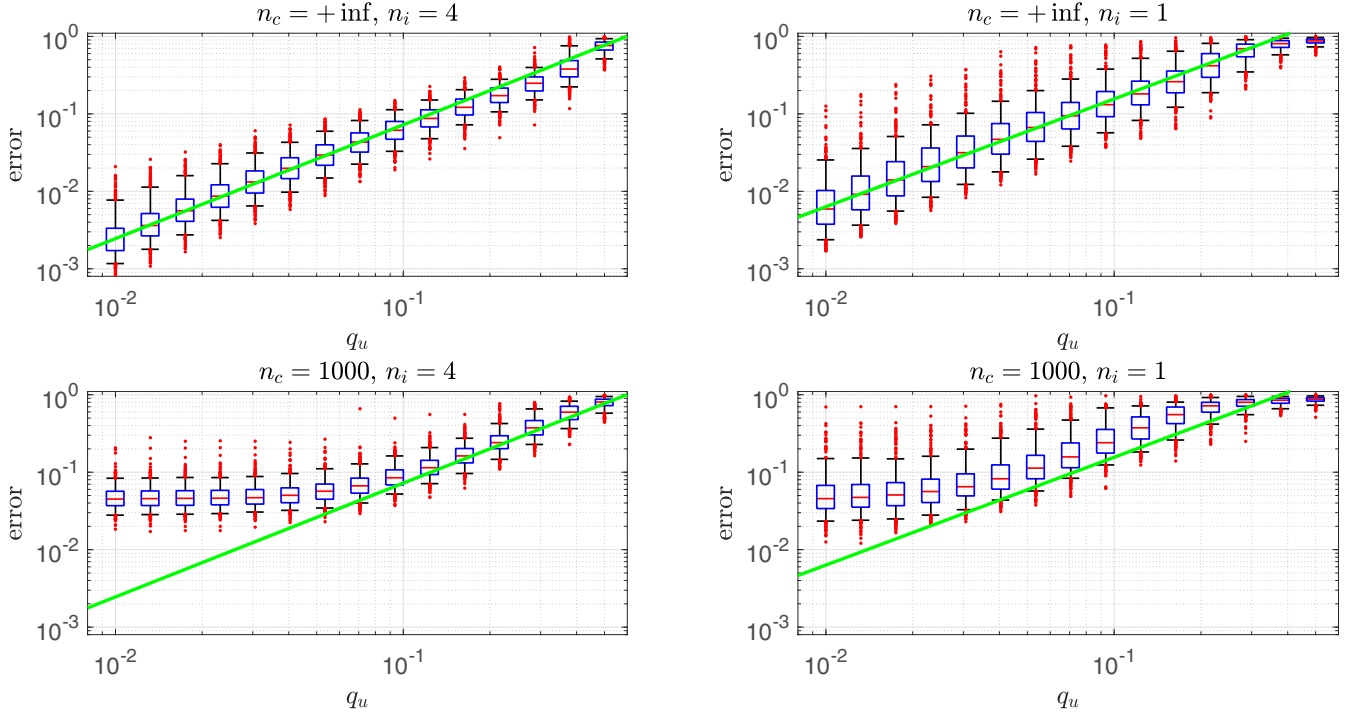


FIG. 13. Box plots of the QPT error with nonunitary processes; the nonunitarity is quantified by q_u . The setup of Fig. 3 ($n_i = 4$) and that of Fig. 4 ($n_i = 1$) are used, with finite and infinite n_c (n_{tot} remains constant for finite n_c). The green lines are computed in the upper plots (with a linear regression on the medians of the errors associated with $q_u < 0.1$ in the log-log plot) with $n_c = +\infty$; they are then duplicated in the lower plots.

are eigenvectors of \mathbf{C} (because any basis that diagonalizes $\mathbf{x}_\ell \mathbf{x}_\ell^*$ contains \mathbf{x}_ℓ up to a global phase). Let us call $e^{i\lambda_\ell}$ the eigenvalue of \mathbf{C} associated with \mathbf{x}_ℓ (they are of unit modulus because \mathbf{C} is unitary).

Let us consider two indices ℓ_1 and ℓ_2 , and let us show that $\mathbf{x}_{\ell_1} \not\perp \mathbf{x}_{\ell_2} \Rightarrow e^{i\lambda_{\ell_1}} = e^{i\lambda_{\ell_2}}$: $\mathbf{x}_{\ell_1}^* \mathbf{x}_{\ell_2} = (\mathbf{C}\mathbf{x}_{\ell_1})^* \mathbf{C}\mathbf{x}_{\ell_2} = e^{i(\lambda_{\ell_2} - \lambda_{\ell_1})} \mathbf{x}_{\ell_1}^* \mathbf{x}_{\ell_2}$. If $\mathbf{x}_{\ell_1} \not\perp \mathbf{x}_{\ell_2}$, then we can divide both sides of the equation by $\mathbf{x}_{\ell_1}^* \mathbf{x}_{\ell_2}$ and we have $1 = e^{i(\lambda_{\ell_2} - \lambda_{\ell_1})} \Rightarrow e^{i\lambda_{\ell_1}} = e^{i\lambda_{\ell_2}}$.

Let ℓ_1 be in $\{1, \dots, n_x\}$. \mathbf{x}_{ℓ_1} is not orthogonal to any column of $\mathbf{F}_S(\mathbf{x}_{\ell_1})$ (by definition of \mathbf{F}_S), and the columns of $\mathbf{F}_S(\mathbf{x}_{\ell_1})$ are also in $\{\mathbf{x}_\ell\}_\ell$. Thus, all columns of $\mathbf{F}_S(\mathbf{x}_{\ell_1})$ have $e^{i\lambda_{\ell_1}}$ as the associated eigenvalue. The same is true for the columns of $\mathbf{F}_S^k(\mathbf{x}_{\ell_1})$ for any $k \geq 1$ (straightforward by mathematical induction). In particular, all columns of $\mathbf{F}_S^{n_x}(\mathbf{x}_{\ell_1})$ have the same associated eigenvalue: $e^{i\lambda_{\ell_1}}$. But (17) guarantees that $\mathbf{F}_S^{n_x}(\mathbf{x}_{\ell_1})$ has rank d . Thus there are d linearly independent columns of $\mathbf{F}_S^{n_x}(\mathbf{x}_{\ell_1})$ that form a basis, they are also eigenvectors of \mathbf{C} [like all columns of $\mathbf{F}_S(\mathbf{x}_{\ell_1})$]. This basis is therefore an eigenbasis of \mathbf{C} , and there is only one associated eigenvalue: $e^{i\lambda_{\ell_1}}$. This means that $\mathbf{C} = e^{i\lambda_{\ell_1}} \mathbf{I}_d$.

APPENDIX C: DECOHERENCE

The systematic errors are not the only types of errors that can occur for the measured states with our QPT setup. Non-systematic errors are assumed to be centered errors; they are not the same for each prepared copy of the measured states. They occur in the following cases:

(1) The preparation of the original states ($|0\rangle$), which is not the same for each copy.

(2) The state-preparation gates (Hadamard gates in Fig. 3) do not have the same behavior for each copy, thus introducing decoherence during the state-preparation process.

(3) Decoherence occurs after the initial state-preparation process.

All those causes of errors, except the last one, can be modeled by considering that the initial states are mixed states. The last cause of error occurs only if the process to be identified is nonunitary (decoherence is nonunitary by nature; if it occurs after the initial states preparation, then it was introduced by the process to be identified). It will be tested separately.

To model the first two types of errors, we simulate mixed input states. A mixed state is parameterized by a positive Hermitian matrix with unit trace $\rho \in \mathbb{H}_d^+(\mathbb{C})$ whose spectral decomposition is $\rho = \sum_{k=1}^d p_k^\rho \mathbf{v}_k^\rho \mathbf{v}_k^{\rho*}$, where the $\{p_k\}$ are the eigenvalues (in decreasing order), they are non-negative and sum to one, and the $\{\mathbf{v}_k\}$ are the eigenvectors. Let us define $q_1 = 1 - p_1^\rho$; it quantifies how close ρ is to a pure state. If $q_1 = 0$, then the initial state is pure and can be represented by the vector \mathbf{v}_1^ρ . With our conventions, q_1 cannot be greater than $1 - \frac{1}{d}$.

We simulate 16 different values of q_1 from 10^{-2} to 0.5. The smaller eigenvalues are all set to $\frac{1-q_1}{d-1}$. We test the performance of our QPT algorithm with those mixed states with and without the “multinomial” errors (with $n_c = 1000$ and $n_c = +\infty$). Those tests are performed with the setup of Fig. 3, with $n_{qb} = 2$ and with random quantum processes. The resulting errors are represented in Fig. 12.

The (green) straight lines from the bottom left to the top right of Fig. 12 are the same for the two graphs. Their slopes and intercepts are computed with a linear regression to fit the medians of the errors associated with $q_1 < 0.1$ and with no systematic error ($n_c = +\infty$). The slope found by the linear regression is 0.9. This means that the median of the error is roughly proportional to $q_0^{0.9}$ if $n_c = +\infty$. This value slightly changes when we change the model of the mixed states in the simulation (it depends on the rank of ρ and on its eigenvalues) but stays roughly the same, slightly below 1.

For the chosen value of n_c ($n_c = 1000$), the ‘‘multinomial’’ error makes the impact of the error generated by the nonpurity of the states negligible as long as $q_0 < 0.05$.

Let us now apply our QPT algorithms to a nonunitary process that introduces decoherence after the initialization. For the sake of simplicity, we use a nonunitary processes (ϵ) in the space of complex matrices ρ modeled as follows:

$$\epsilon(\rho) = p_u \mathbf{M} \rho \mathbf{M}^* + q_u \text{tr}(\rho) \mathbf{I}_d, \quad (\text{C1})$$

where \mathbf{M} is the unitary matrix that represents the unitary part of the process to be identified, and p_u and q_u are real numbers between 0 and 1 such that $p_u = 1 - q_u$. We will vary q_u . Like all quantum processes, ϵ is linear and preserves the positivity and the trace of Hermitian matrices. For each value of q_u , we test our algorithm with the setups of Fig. 3 and Fig. 4 [30] for 500 nonunitary processes (corresponding to randomly generated \mathbf{M}). Then the error (26) between \mathbf{M} and the estimated $\widehat{\mathbf{M}}_{LS}$ is computed (we consider that \mathbf{M} represents the process, and that the nonunitary part of ϵ is noise).

We simulate this error with the setup of Fig. 3 with $n_{qb} = 2$, $n_i = 4$, $n_s = 2$ (like for almost all previous tests) and with the setup of Fig. 4 with $n_i = 1$ and $n_s = 5$. It is relevant to test the latter setup here because the nonunitary process generates more decoherence if the state is observed during a longer period (higher n_s). Therefore, the error should be greater with $n_s = 5$ than with $n_s = 2$.

The QPT error is displayed in Fig. 13 with and without a multinomial error and with $n_i = 4$ (implicitly with the setup of Fig. 3, $n_s = 2$) and $n_i = 1$ (implicitly with the setup of Fig. 4, $n_s = 4$). To compare fairly the two setups corresponding to $n_i = 1$ and $n_i = 4$, the total number of prepared initial states $n_{i\alpha} = n_i n_s n_c$ has to be the same for both setups. Therefore, with a multinomial error, we set $n_c = 1000$ for the setup of Fig. 3 ($n_i = 1$, $n_s = 5$), and we set $n_c = 1600$ for the setup of Fig. 4 ($n_i = 4$, $n_s = 2$). Like with mixed input states, the (green) lines from bottom left to top right are computed with a linear regression to fit the medians of the errors corresponding to $q_u < 0.1$ with $n_c = +\infty$. Two sets of slopes and intercepts

for the green lines are computed on the upper two graphs, and they are replicated in the lower graphs.

As expected, the configuration of Fig. 3 ($n_i = 4$) is more adapted than the setup of Fig. 3 ($n_i = 1$) to nonunitary processes. In both cases, for $n_c = +\infty$, the medians of the errors are roughly proportional to $q_u^{1.5}$ (the slopes of the green lines are roughly 1.5) for reasonable values of q_u ($q_u < 0.1$), but the proportionality coefficient is much higher with $n_i = 1$ than with $n_i = 4$ (it is roughly 2.7 times higher). The difference between the two setups is less obvious with a multinomial error (finite n_c), but it remains noticeable. With the values of n_c we chose, the multinomial error makes the impact of the error generated by the nonunitarity of the process negligible when $q_u < 0.05$ for $n_i = 4$, and when $q_u < 0.02$ for $n_i = 1$.

APPENDIX D: TABLE OF THE MEASUREMENT COUNTS

The following Table II contains all the results obtained from the cloud computing platform that are necessary to replicate our results.

TABLE II. Measurement counts on the actual quantum computer. For example, the value 243 in the row called ZZ 00 and the column called $\mathbf{M}v_1$ means that when measuring $\mathbf{M}v_1$ with measurement type ZZ we obtained the first outcome (called 00 here) 243 times.

	$\mathbf{M}v_1$	$\mathbf{M}v_2$	$\mathbf{M}v_3$	$\mathbf{M}v_4$	\mathbf{M}^2v_1	\mathbf{M}^2v_2	\mathbf{M}^2v_3	\mathbf{M}^2v_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

[1] I. L. Chuang and M. A. Nielsen, *J. Mod. Opt.* **44**, 2455 (1997).
[2] J. F. Poyatos, J. I. Cirac, and P. Zoller, *Phys. Rev. Lett.* **78**, 390 (1997).
[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
[4] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, *Phys. Rev. A* **77**, 032322 (2008).

[5] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White, *Phys. Rev. Lett.* **106**, 100401 (2011).
[6] A. Kalev, R. L. Kosut, and I. H. Deutsch, *npj Quantum Inf.* **1**, 15018 (2015).
[7] A. Smith, C. A. Riofrío, B. E. Anderson, H. Sosa-Martinez, I. H. Deutsch, and P. S. Jessen, *Phys. Rev. A* **87**, 030102(R) (2013).
[8] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, *Phys. Rev. Lett.* **105**, 150401 (2010).

- [9] C. H. Baldwin, A. Kaley, and I. H. Deutsch, *Phys. Rev. A* **90**, 012110 (2014).
- [10] D. M. Reich, G. Gualdi, and C. P. Koch, *Phys. Rev. A* **88**, 042309 (2013).
- [11] S. Kimmel, G. H. Low, and T. J. Yoder, *Phys. Rev. A* **92**, 062315 (2015).
- [12] A. C. Keith, C. H. Baldwin, S. Glancy, and E. Knill, *Phys. Rev. A* **98**, 042318 (2018).
- [13] M. B. Ruskai, S. Szarek, and E. Werner, *Linear Algebra Appl.* **347**, 159 (2002).
- [14] F. Verdeil and Y. Deville, *Phys. Rev. A* **107**, 012408 (2023).
- [15] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, *Phys. Rev. A* **87**, 062119 (2013).
- [16] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout, *Quantum* **5**, 557 (2021).
- [17] Y. Deville and A. Deville, in *International Conference on Latent Variable Analysis and Signal Separation* (Springer, 2015), pp. 184–192.
- [18] Y. Deville and A. Deville, *IFAC-PapersOnLine* **50**, 11731 (2017).
- [19] Y. Deville and A. Deville, *Phys. Rev. A* **101**, 042332 (2020).
- [20] This requires the preparation procedure to be known and reproducible so that several copies of each used state may be prepared. It is not a violation of the no cloning theorem: the latter does not apply if we prepared the state that we want to reproduce.
- [21] F. Verdeil, Y. Deville, and A. Deville, in *2021 IEEE Statistical Signal Processing Workshop (SSP)* (IEEE, Rio de Janeiro, 2021), pp. 161–165.
- [22] F. Verdeil and Y. Deville, in *Physical Sciences Forum*, Vol. 5 (Multidisciplinary Digital Publishing Institute, 2022), p. 29.
- [23] D. Goyeneche, G. Cañas, S. Etcheverry, E. S. Gómez, G. B. Xavier, G. Lima, and A. Delgado, *Phys. Rev. Lett.* **115**, 090401 (2015).
- [24] Y. Chen, M. Farahzad, S. Yoo, and T.-C. Wei, *Phys. Rev. A* **100**, 052315 (2019).
- [25] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta, *Phys. Rev. A* **103**, 042605 (2021).
- [26] K. Arun, *SIAM J. Matrix Anal. Appl.* **13**, 729 (1992).
- [27] G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 2013).
- [28] `Arn:aws:braket::device/qpu/ionq/ionQdevice`.
- [29] R. A. Horn and C. R. Johnson, *Matrix Analysis* (Cambridge University Press, 2012).
- [30] Those figures have been designed for unitary processes. However, we can imagine that the process represented by \mathbf{M} is not unitary. In that case (considered here), the input states (\mathbf{v}_k) are pure, but the output states [which should be called $\epsilon(\mathbf{v}_k \mathbf{v}_k^*)$ but are called $\mathbf{M}\mathbf{v}_k$] are mixed states.