Discover Quantum Science

RESEARCH Open Access

Multi-stage tomography based on eigenanalysis for high-dimensional dense unitary processes in gate-based quantum computers

Yannick Deville[1*] and Alain Deville[2]

*Correspondence:
Yannick Deville
yannick.deville@irap.omp.eu
[1]IRAP (Institut de Recherche en Astrophysique et Planétologie), Université de Toulouse, CNRS, CNES, 31400 Toulouse, France
[2]IM2NP UMR 7334, Aix-Marseille Université, CNRS, 13397 Marseille, France

Abstract

A major quantum information processing tool is Quantum Process Tomography (QPT), which consists of methods that aim at identifying, i.e. estimating, a quantum process. QPT may especially be applied to quantum gates, that are widely used as the building blocks of quantum computers, in order to experimentally characterize their actual behavior. In this paper, we address unitary processes (corresponding to isolated systems), that may be dense, i.e. that do not have sparsity constraints. The proposed approaches to QPT require one to first perform Quantum State Tomography (QST). Afterwards, the core of our QPT methods applies to a significant number of qubits and thus to a high state space dimension and to complex problems. Our methods first achieve part of QPT by performing an eigenanalysis of the density matrix of a process output estimated with QST, by taking advantage of the unitarity of the process. Our first resulting class of complete algorithms uses only one such eigendecomposition and is thus "single-stage". We then extend our approach to algorithms that are "multiple-stage", i.e. that perform several eigendecompositions. This allows one to handle high-dimensional state spaces with a reduced sensitivity to the estimation errors that result from the use of an arbitrary QST algorithm as a preprocessing step of our QPT methods. Our multi-stage methods first include two-stage versions and then dichotomic versions, with a number of stages that increases with the considered state space dimension. Simulations were successively performed with different magnitudes for QST errors / noise. They provide evidence of the relevance of all proposed methods. Their single-stage and two-stage versions are efficient up to 13 qubits when executed on a very simple platform: an 8-year old PC with only 16 GB of RAM. An even higher accuracy (Normalized Mean Square Error down to 10^{-3}) is obtained with multi-stage methods.

Keywords Eigenanalysis of density matrix, High-dimensional quantum state space, Quantum gate characterization, Quantum machine learning (supervised/semi-supervised/unsupervised), Quantum process tomography, Unitary quantm gate

1 Introduction

Quantum process tomography (QPT) may be considered as the quantum counterpart of classical system identification [1], that is, system parameter estimation. It was mainly[1] introduced in 1997 in [2] and then extended (together with quantum process learning) e.g. in [1, 3–27]. It has been applied both to unitary processes (corresponding to isolated systems) and to non-unitary processes (corresponding to open systems). It is a major quantum data processing tool, since it especially allows one to characterize the actual behavior of quantum gates, that may be used as the building blocks of quantum computers. This gate-based (or circuit-based) approach builds upon the unitary transforms performed by quantum gates (see Chapter 4 of [1]). It is currently one of the most prominent quantum computing paradigms and various commercial products based on this paradigm are available. This e.g. includes IBM's Qiskit environment (see [28–30] and especially [31] for the use of reversible unitary gates), Amazon's Braket tool (see [32] especially for reversible transformations), and Mathworks's software package (see [33] e.g. for their statement "Quantum gates represent reversible operations that transform the quantum state according to unitary matrices") . Due to the importance of the abovementioned *unitary* processes, various methods dedicated to such processes have been proposed, for performing unitary quantum process estimation/learning (see e.g. [3, 13, 22–27]) and e.g. for the related topic of the inversion and transposition of unknown unitary operations [34]. Besides, the recent review article [35] on "Challenges and opportunities in quantum optimization" not only refers to the use of unitary processes, but also comments about the quantum imaginary time evolution (QITE) approach by stating: "because QITE is a non-unitary evolution, it is challenging to implement on a quantum computer, and only heuristic variants of QITE exist at present. " This indirectly shows the importance of unitary operations in quantum computing.

QPT is a "'data-driven" approach, and can thus be considered as a type of quantum machine learning (QML) problem (see [36] for a taxonomy of QML problems), because the input–output transform performed by the considered process is generally inferred from the values of input quantum states and/or from measurement results for output quantum states of that process. More precisely, usual QPT methods require one to fix a priori and hence know the above input "values" (i.e. states) of the process and to use information about the corresponding output "values" (i.e. states), that is derived from measurements performed at the output of the considered process. This corresponds to the supervised version of machine learning, also called the non-blind version of machine learning [37]. In contrast, the other extreme case is the one which requires the most limited knowledge about the process input whereas output measurements are still used, and this corresponds to the unsupervised, or blind, QPT (BQPT) configuration [37]. BQPT methods were introduced in [38] and then especially developed in [37, 39, 40]. In that case, the individual input state values are completely unknown, that is, neither fixed a priori nor estimated by means of measurements: in [38, 39], the input consists of unknown unentangled deterministic-coefficients pure states (DCPS); in [37, 40], it consists of random-coefficient pure states (RCPS) and the information available about these RCPS is restricted to some of their global, i.e. statistical, properties. DCPS are the usual

[1] See also [1] p. 398 for the other earliest references.

form of pure states of quantum physics, whereas RCPS were introduced in [41] and then discussed in various related papers, especially including [42].

Between the above two extreme cases, a recently introduced approach (see [22] and references therein) was claimed to be semi-blind (i.e. semi-supervised), because it does not require the values of the input quantum states to be fixed a priori, but it estimates them, by also performing measurements at the input of the process for part of the available copies of these input states. This method thus allows one to use almost any values for the input states, whereas various supervised QPT methods are more constraining because they are only applicable to a specific, predefined, set of input values.

As detailed below, the different parts and variants of the QPT algorithms proposed in the present paper cover all above three types of learning. Some of them require their input states to be known, so that these states may be fixed a priori (supervised learning) or estimated from measurements (semi-supervised learning). Other parts and variants of our algorithms use input states whose values are not known, but that are required to meet a given condition (unsupervised learning). Besides, part of the proposed algorithms use "pure states" , where we employ the term "pure states" in the above-defined usual sense, that is, DCPS. In contrast, other proposed algorithms use statistical mixtures.

Within the above framework, we focus on an original case which combines the following two features. First, the considered process is constrained to be unitary, which corresponds to addressing a system that is isolated from its environment. Second, we are mainly interested in the case when the process has a high dimension, because it involves a significant number of qubits . This allows one to tackle more complex problems. We stress that we do not set any other conditions on the considered process. In particular, we allow (but do not request) the matrix that represents that process in the considered basis to be a so-called dense, i.e. non-sparse, matrix (we here use the standard terminology: a sparse matrix is a matrix that has a large percentage of zeros; see e.g. [43]). In contrast, some approaches from the literature for QPT and the related topic of Hamiltonian estimation request sparsity (see e.g. [12, 44]) , but this was considered to be less relevant than unitarity in [22] and we explained above when the unitarity assumption is relevant.

Concerning the "mathematical principles" of the proposed QPT methods, in this paper we show that one of the key ingredients of our approach is the eigendecomposition of the (estimated) density matrix of a mixed output state of the considered process. Before moving to technical details and a variety of required extensions of this idea in the subsequent sections, this major idea can here be outlined as follows. The first output density matrix of the considered process is here denoted as ρ_2. As discussed in Sect. 2, this density matrix reads

$$\rho_2 = U \; \rho_1 \; U^\dagger \tag{1}$$

where U is the unitary matrix that represents the process to be estimated in the considered basis, $.^\dagger$ is the transconjugate (i.e. Hermitian transpose) and ρ_1 is the density matrix of the first input mixed state that yields the output state ρ_2. The idea is then that, when the density matrix ρ_1 is selected to be diagonal in the considered and fixed basis (computational basis), Eq. (1) is an eigendecomposition of ρ_2. Therefore, in practice, starting from an estimate of ρ_2 derived from measurements, its eigendecomposition may be hoped to yield a result directly related to an estimate of U. This idea will then have to be refined further in this paper because: 1) eigendecomposition yields indeterminacies

(defined below) and 2) fluctuations in output density matrix estimation limit the performance of this preliminary approach and then lead us to extend it in this paper.

The above preliminary description shows that the type of approaches proposed in this paper has some connections with works that were previously reported both in the classical and quantum domains. The connections with classical approaches are discussed in section 1 of the Supplementary Document, whereas those with the quantum domain may be defined as follows. A few recent papers dealing with quantum data also consider (1) in the framework of QPT. However, they only have the following limited relationships with our investigation. Ref. [45] aims at deriving the number (and values) of quantum states that should be applied to the input of a quantum process to be able to identify that process. However, that paper does not focus on the same input values as in the present paper and, as also noted in [22], it does not provide an explicit QPT algorithm. Ref. [3] provides such algorithms, that are different from ours and that are tested only for a qudit corresponding to a Hilbert space dimension $d = 5$, whereas we consider dimensions up to $d = 2^{13} \simeq 8000$.

QPT algorithms may also be split in two classes, from the following point of view. Part of them first use the results of measurements at the output of the system by explicitly performing quantum state tomography (QST) . This means that they estimate the ket (for a pure state) or the density matrix (for a statistical mixture) that represents that ouput state. They then use these estimated output states (and the corresponding input states, their estimates or their properties) so as to infer the considered process. Other QPT methods instead directly use the results of the considered types of measurements. In the present paper, we consider the first class of QPT methods and we focus on the part dedicated to QPT itself, as opposed to QST, as detailed in Sect. 5. This also means that the complexity of our methods directly results from that of the considered QST algorithms. This topic is therefore addressed in more detail in Sect. 5, when discussing QST. In contrast, complexity analysis may take different forms for other types of algorithms: see the discussion provided in [45] about the required number of input states.

The remainder of this paper is organized as follows. Sect. 2 is devoted to the introduction of several variants of our basic type of QPT methods, that is, single-stage methods. These methods have a limitation, that is defined in Sect. 3. Therefore, we first introduce the two-stage extensions of the above methods in Sect. 3, and then their general multi-stage form in Sect. 4. Numerical results illustrate the performance of the proposed methods in Sect. 5 and conclusions are drawn from this investigation in Sect. 6.

2 Proposed single-stage QPT methods

As stated above, in all the configurations considered in this paper, the quantum system is assumed to be isolated from its environment (the proposed approaches may therefore also be used as an approximation for systems that have a limited interaction with their environment and that therefore lead to near-unitary processes, as in [3]). It is well-known that, in that case, the temporal evolution of the state of that system between given initial and final times is governed by a unitary operator (see [1, 46]; see also [47, 48] for the relationship with the system's Hamiltonian). More precisely, using the above-defined notations, we consider a (first) mixed state defined by a density matrix ρ_1 at an initial time. As explained e.g. in [1, 46], the state of that system at a subsequent, final,

time is then defined by a density matrix ρ_2 that may be expressed as in (1), where U is a unitary matrix, that depends on the dynamics of the considered system.

The algorithms proposed in this section to estimate U then consist of two parts. Each part involves a given type of input quantum state and an algorithm for exploiting the associated output quantum state (and possibly the input state). Each of the following subsections describes one of these two parts.

2.1 First part of the methods

We here consider the situation when the first state applied to the analyzed process is a given statistical mixture, represented by the density matrix ρ_1. The process output is then also a mixed state and its density matrix ρ_2 may be expressed as (1). More specifically, we here introduce a first variant, in which the matrix ρ_1 is diagonal . Section 3 of the Supplementary Document addresses the more general case when this constraint is removed.

It should be noted that, when one considers that one is able to create a pure state, then, creating a mixed state with an input density matrix ρ_1 that is diagonal is not a problem, as will now be shown. We here consider the approach initially used by von Neumann in [49] to define a mixed state, i.e. a statistical mixture: such a state is expressed as being composed of a set of pure states, each of them being associated with a probability of occurrence. Then, to create experimental data corresponding to the state defined by the diagonal density matrix ρ_1, one just has to repeatedly, randomly, select one of the pure states that form the considered basis of the state space, moreover selecting the kth basis state with a probability that is equal to the k th value on the diagonal of ρ_1. This simplicity of implementation is worth noting because, in contrast, the authors of [3, 45] consider that using a mixed input state is too complex (maybe because they have non-diagonal density matrices in mind), so that they choose to restrict themselves to pure states, which has the drawback of requesting many such multiqubit input states, as opposed to only two multiqubit input states in the complete approach that we here start to describe (namely the mixed state with a density matrix ρ_1 considered here, and then the pure state $|\Psi_1\rangle$ in Sect. 2.2). Beyond their relationship based on considering (1), our approach and the one in [45] are therefore quite different.

The matrix ρ_2 is known to be Hermitian, since it is a density matrix (see e.g. [46] p. 283). Therefore, it is normal (see [50] p. 100[2]) and hence unitarily diagonalizable: see Theorem 1 and comments in its proof (all theorems cited in the core of this paper are provided in Appendix A). The first part of the QPT method proposed below is based on the eigendecomposition of ρ_2 (because (1) is such an eigendecomposition, as detailed below). More precisely, in practice, what is initially available for performing an eigendecomposition is not the theoretical matrix ρ_2 but an estimate $\widehat{\rho}_2$ of ρ_2, provided by a QST algorithm (we here use standard notations, e.g. as in [52–54]: an estimate of an unknown, deterministic, scalar, vector or matrix θ is denoted as $\widehat{\theta}$). Therefore, our QPT method preferably starts with two preprocessing steps, that consist of first deriving a Hermitian estimate $\widehat{\rho}_3$ from $\widehat{\rho}_2$ and then a unit-trace estimate $\widehat{\rho}_4$ from $\widehat{\rho}_3$ (see details in Appendix B), unless the considered QST algorithm already guarantees that $\widehat{\rho}_2$ meets these two properties.

[2] Throughout this paper, the reader may refer to [51], instead of [50].

Then, the core of our QPT algorithm performs an eigendecomposition of the Hermitian matrix $\widehat{\rho}_4$. This eigendecomposition problem has several solutions, corresponding to the indeterminacies of eigendecomposition detailed below. Any of these solutions is defined by two matrices, hereafter denoted as D_4 and V_4. D_4 is a diagonal matrix that contains the eigenvalues of $\widehat{\rho}_4$ in an arbitrary order (see e.g. [50] p. 102). Moreover, each column of V_4 is an eigenvector of $\widehat{\rho}_4$ (see (2) and Theorem 2) and V_4 is here unitary (because $\widehat{\rho}_4$ is unitarily diagonalizable), hence $V_4^{-1} = V_4^{\dagger}$. These results V_4 and D_4 of the eigendecomposition are linked to the initial matrix $\widehat{\rho}_4$ as follows:

$$\widehat{\rho}_4 = V_4 \; D_4 \; V_4^{-1} \tag{2}$$

$$= V_4 \; D_4 \; V_4^{\dagger}. \tag{3}$$

Comparing (3) to (1), with $\widehat{\rho}_4$ an estimate of ρ_2 and with ρ_1 a diagonal matrix, shows that (1) is one of the possible eigendecompositions of $\widehat{\rho}_4$ up to estimation errors. Therefore, the matrices D_4 and V_4 obtained in practice with our algorithm are respectively equal to ρ_1 and U up to the indeterminacies of the eigendecomposition process (and up to estimation errors). In particular, V_4 is the first obtained quantity related to the estimation of U (it could therefore be denoted as $\widehat{U}_1$). We now define its indeterminacies in more detail, and we show how to solve them.

In our first type of QPT methods described here, we constrain all diagonal values of the user-defined matrix ρ_1 to be different. This means that all eigenvalues in the eigendecompositions of ρ_2 and hence $\widehat{\rho}_4$ (ignoring estimation errors) are different. Therefore, each of them is associated with a different one-dimensional eigenspace . In other words, all eigenvectors corresponding to a given eigenvalue are along the same direction and the directions associated with two different eigenvalues are different (this is true for any complex square matrix: see [50] p. 47) and even orthogonal: see Theorem 3. Each eigenvalue is thus coupled with a single eigenvector in any considered eigendecomposition. The first indeterminacy of the eigendecomposition of $\widehat{\rho}_4$ is then a possible *coupled* permutation of 1) the columns of the matrix V_4 obtained as a first quantity towards the estimation of U in this eigendecomposition and 2) the values along the diagonal of the matrix D_4, due to the above-mentioned arbitrary order of the latter diagonal values. This permutation problem may be solved as follows. As stated above, we constrain all diagonal values of ρ_1 to be different (at this stage of the discussion, we restrict ourselves to a theoretical approach, where we only need the eigenvalues to be different; further in this paper, we will discuss the gap required between them, when considering practical implementations of our approach : see Appendix C for the method considered in the present section ; moreover, the practical consequences of this limited gap directly appear in the resulting performance obtained in the tests reported in Sect. 5). Moreover, we constrain all diagonal values of ρ_1 to be placed on that diagonal according to a given order. Hereafter, we will consider the case when they are in decreasing order. With respect to the above-defined machine learning terminology, this part of our algorithm is thus unsupervised: it does not require the exact value of the input state, i.e. the input density matrix ρ_1, to be known but it sets limited conditions on some of its properties.

The above known order of the diagonal values of ρ_1 is used to postprocess D_4, by permuting the values obtained on its diagonal so that they are in the same order as the order that we impose on ρ_1. More precisely, in the case considered hereafter, we permute the

values obtained on the diagonal of D_4 so that they are in decreasing order (if some eigenvalues obtained in practice are nonreal, they are ordered according to their moduli). Since we now know which permutation is required to this end, we then use that specific permutation to reorder the columns of V_4 in the same way. The matrix with reordered columns thus obtained is denoted as $\widehat{U}_2$, since it is the second quantity that we build for estimating U.

Thanks to the above approach, the only indeterminacy that remains in $\widehat{U}_2$ with respect to U is an unknown complex scale factor in each of its columns. These factors are handled as detailed in Appendix D and this yields the modified version of $\widehat{U}_2$ that reads

$$\widehat{U}_2 = \left[e^{i\phi_1}u_1, \ldots, e^{i\phi_d}u_d\right] \tag{4}$$

where $u_1, \ldots, u_d$ are the column vectors of the actual matrix U and $\phi_1, \ldots, \phi_d$ are the phase indeterminacies with respect to the above vectors that remain at this stage. Denoting D_ϕ the diagonal matrix that contains the unknown phase factors $e^{i\phi_1}, \ldots, e^{i\phi_d}$ on its main diagonal, (4) reads

$$\widehat{U}_2 = UD_\phi. \tag{5}$$

The second part of our algorithm, presented below in Sect. 2.2, aims at determining all phase factors $e^{i\phi_1}, \ldots, e^{i\phi_d}$, up to a global phase factor (a process matrix U can only be defined up to a single, i.e. global, phase factor).

2.2 Second part of the methods

Thanks to the limited number of unknowns (namely the phase factors $e^{i\phi_1}, \ldots, e^{i\phi_d}$) that remain in the matrix $\widehat{U}_2$ of (4) obtained in Sect. 2.1, a single and pure state is then sufficient for estimating these unknowns. This yields the first variant of the second part of our methods, presented here. However, one may instead use a mixed state for estimating these unknowns, as shown in Sect. 4 of the Supplementary Document.

We therefore here apply a known pure state, defined by a ket $|\Psi_1\rangle$, at the input of the process (this part of the algorithm is therefore supervised, or semi-supervised if $|\Psi_1\rangle$ is initially unknown but estimated by performing measurements for part of the available copies of that state ; for the supervised case, Appendix E explains how $|\Psi_1\rangle$ is chosen). The corresponding pure state at the output of that process is defined by the ket

$$|\Psi_2\rangle = U|\Psi_1\rangle. \tag{6}$$

What is available in practice is a ket $|\widehat{\Psi}_2\rangle$, obtained by performing a QST at the output of the process, and equal to $|\Psi_2\rangle$ up to an unknown phase factor and to estimation errors (see again Appendix B concerning a possible normalization of the QST output used here). When neglecting the above estimation errors,

$$|\widehat{\Psi}_2\rangle = e^{i\theta}|\Psi_2\rangle \tag{7}$$

$$= e^{i\theta}U|\Psi_1\rangle. \tag{8}$$

Our algorithm computes the ket

$$|\Psi_3\rangle = \widehat{U}_2^\dagger|\widehat{\Psi}_2\rangle, \tag{9}$$

based on the following motivation: combining (9, 5, 8) and considering that U is unitary (hence $U^\dagger U = I$) yields

$$|\Psi_3\rangle = e^{i\theta} D_\phi^* |\Psi_1\rangle \tag{10}$$

where * stands for conjugation. This equation is very useful because, apart from the global phase θ, its only unknown is D_ϕ, i.e. the set of phase factors $e^{i\phi_1}, \ldots, e^{i\phi_d}$ to be determined. This problem is easily solved by considering each component k, denoted as $|.\rangle_k$, of the above kets. Using a ket $|\Psi_1\rangle$ with nonzero components, (10) yields

$$\frac{|\Psi_3\rangle_k}{|\Psi_1\rangle_k} = e^{i(\theta - \phi_k)} \qquad \forall k \in \{1, \ldots, d\}. \tag{11}$$

The associated matrix form reads

$$\mathrm{diag}(|\Psi_3\rangle \oslash |\Psi_1\rangle) = e^{i\theta} D_\phi^* \tag{12}$$

where $\oslash$ is the element-wise division for vectors (as defined in Appendix A) and $\mathrm{diag}(v)$ is the diagonal matrix that contains all elements of a vector v on its main diagonal.

Based on the above properties, the final step of our algorithm creates, as follows, a new estimate of U denoted as $\widehat{U}_5$ (the intermediate notations $\widehat{U}_3$ and $\widehat{U}_4$ are not used here but are required for the two-stage methods of Sect. 3):

$$\widehat{U}_5 = \widehat{U}_2 \ \mathrm{diag}(|\Psi_3\rangle \oslash |\Psi_1\rangle). \tag{13}$$

This approach is used because, due to (12 and 5), Eq. (13) yields

$$\widehat{U}_5 = e^{i\theta} U \tag{14}$$

which means that $\widehat{U}_5$ succeeds in restoring U, up to a global phase factor (and again up to the estimation errors that were mentioned above but implicit in the above equations).

A pseudo-code of the version of our algorithm corresponding to all this Sect. 2 is provided in Algorithm 1. This algorithm is called EQPT1, because it is our first proposed Eigenanalysis-based QPT algorithm.

Input : a) Estimate $\widehat{\rho}_4$ of output density matrix ρ_2 (provided by Quantum State Tomography (QST) and obtained for input density matrix (C7)). b) Estimate $|\widehat{\Psi}_2\rangle$ of output ket $|\Psi_2\rangle$ (provided by QST and obtained for all components of input ket $|\Psi_1\rangle$ equal to $1/\sqrt{d}$).

Output: Estimate $\widehat{U}_5$ of quantum process matrix U.

begin

`/* Exploit` $\widehat{\rho}_4$`: */`

Eigendecomposition: derive 1) a diagonal matrix D_4 that contains the eigenvalues of $\widehat{\rho}_4$ in an arbitrary order and 2) a matrix V_4 whose columns are eigenvectors of $\widehat{\rho}_4$ in the same order as eigenvalues;

Reorder the eigenvalues in D_4 in decreasing order and apply the same permutation to the columns of V_4 to create the matrix $\widehat{U}_2$;

`/* If the above eigendecomposition algorithm does not yield` *`unit-norm`* `eigenvectors, then create them as follows: */`

Divide each column of $\widehat{U}_2$ by its norm;

`/* Exploit` $|\widehat{\Psi}_2\rangle$`: */`

$|\Psi_3\rangle = \widehat{U}_2^{\dagger}|\widehat{\Psi}_2\rangle$;

$\widehat{U}_5 = \widehat{U}_2\ \mathrm{diag}(|\Psi_3\rangle \oslash |\Psi_1\rangle)$;

end

Algorithm 1 EQPT1: first Eigenanalysis-based Quantum Process Tomography algorithm (composed of one stage). This version is for the case when the QST algorithm provides estimates that meet the known properties of the actual quantities. Otherwise, see Appendix B.

3 Proposed two-stage QPT methods

3.1 Motivation

The dimension d of the state space (and hence the dimension $d \times d$ of the considered process matrix U) that can be addressed by the type of QPT methods proposed in Sect. 2 is limited by the following phenomenon. As explained in that section, these methods require all eigenvalues of the input density matrix ρ_1 of the process to be different. Moreover, they should be separated by a minimum gap to be distinguishable, so that their estimates are not permuted in practice (in order to reorder them and the associated eigenvectors correctly). Besides, all of them range from 0 to 1, because they are nonnegative and their sum is equal to one, as stated above. All these conditions yield an upper bound, denoted as n_{max}, for the acceptable number of eigenvalues of ρ_1. Since the number of (different) eigenvalues of ρ_1 is always equal to d for the methods of Sect. 2, this yields an upper bound for the state space dimension d that can be handled by these methods, and this bound on d is also equal to n_{max}:

$$d \leq n_{max}. \tag{15}$$

In the present section, we aim at addressing higher values of d. To this end, we move to the case when some eigenvalues of the input density matrix ρ_1 are equal and we therefore introduce extensions of the QPT methods of Sect. 2 to be able to handle that more complex case.

3.2 First part of the methods

As in Sect. 2.1, the method proposed here starts by considering a given mixed state, freely selected by the user, that has a diagonal density matrix ρ_1 and that is applied to the input of the considered unitary process defined by a matrix U. However, we here

accept to have identical values on the main diagonal of ρ_1, at the expense of then having to handle more complex eigendecomposition properties for the process output, as shown below. More precisely, each value that appears on the main diagonal of ρ_1 appears d_1 times (where the value of the parameter d_1 of ρ_1 is freely selected by the user), on adjacent rows, and that diagonal contains d_2 different values. The number of rows of ρ_1, equal to the dimension of the considered state space, is thus equal to

$$d = d_1 d_2 \tag{16}$$

(in practice, we aim at considering the case when d_1, d_2, and hence d are powers of 2, with an application to qubits). If requesting the same minimum gap between different eigenvalues as in Sect. 2.1, or in other words, if replacing the bound (15) on d in Sect. 2 by the same bound value for d_2 instead of d here, then the maximum value of the state space dimension (16) that can be handled here is d_1 times higher than the maximum dimension that can be addressed by the methods of Sect. 2, which is our goal, ideally. However, it is not guaranteed that the methods proposed hereafter can handle the same gap between different eigenvalues as in Sect. 2.1 with the same accuracy, especially due to the more complex structure of these extended methods. The practical accuracy of all proposed methods is therefore investigated numerically further in this paper.

Denoting $r_1, \ldots, r_{d_2}$ the different eigenvalues of ρ_1, that chosen density matrix reads

$$\rho_1 = \begin{bmatrix} r_1 & 0 & \ldots & & & & 0 \\ & \ddots & & & & & \\ & & r_1 & & & & \\ & & & r_2 & & & \\ & & & & \ddots & & \\ & & & & & r_2 & \\ & & & & & & \ddots \\ 0 & \ldots & & & & 0 & r_{d_2} \end{bmatrix} \tag{17}$$

with each diagonal value repeated d_1 times. This may also be expressed as

$$\rho_1 = \mathrm{diag}(r) \otimes I_{d_1} \tag{18}$$

where r is the vector that contains the values $r_1, \ldots, r_{d_2}$, whereas I_{d_1} is the $d_1 \times d_1$ identity matrix and $\otimes$ stands for the Kronecker product [55]. Without loss of generality, we here again request the values on the diagonal of ρ_1 to be in nonincreasing order : $r_1 > r_2 > \cdots > r_{d_2}$. Moreover, we preferably use values that are uniformly distributed (that is, multiples of the same step, with zero excluded), as in the method of Sect. 2 (see details for the latter method in Appendix C) , here again in order to maximize the gap between any two adjacent values (the practical consequences of this limited gap here again directly appear in the resulting performance obtained in the tests reported in Sect. 5) . In the configuration of the present Sect. (C7) is thus replaced by:

$$r_k = \frac{2(d_2 - k + 1)}{d(d_2 + 1)} \qquad k \in \{1, \ldots, d_2\}. \tag{19}$$

The density matrix (17) can be created by using exactly the same procedure as described at the beginning of Sect. 2.1, since the latter procedure applies to any diagonal density matrix. So, briefly, here again, in the considered mixed state, each k th basis state is

drawn with a probability that is equal to the k th value on the diagonal of ρ_1. The only specific feature faced here is that some diagonal values of (17), and therefore some probabilities of drawing basis states, are equal.

Here again, the process output corresponding to ρ_1 is a mixed state and its density matrix ρ_2 may be expressed as (1). Moreover, as in Sect. 2.1, ρ_2 is Hermitian and hence unitarily diagonalizable, and it has unit trace. Besides, in practice, an estimate $\widehat{\rho}_2$ of ρ_2 is derived by a QST algorithm and, if required by the considered QST algorithm, we preprocess this estimate of ρ_2 in order to derive its Hermitian and unit-trace version, denoted as $\widehat{\rho}_4$. We then perform an eigendecomposition of $\widehat{\rho}_4$. The analysis of the properties of this eigendecomposition here requires additional care, because the considered matrix has various *identical* eigenvalues.

The properties of Sect. 2.1 that remain true here are as follows. The eigendecomposition is not unique, Eqs. (2, 3) still apply, V_4 and D_4 are respectively a unitary matrix and a diagonal matrix, the k th column of V_4 is an eigenvector of $\widehat{\rho}_4$ and the associated eigenvalue is the k th value on the diagonal of D_4 (see again Theorem 2 for the latter property).

In contrast, what is different here as compared with Sect. 2.1 is that the same eigenvalue here appears d_1 time and is thus associated with d_1 eigenvectors, instead of only one in Sect. 2.1. Eigendecomposition properties are more complex than above in this general framework but, fortunately, their complexity is here limited by the fact that the matrix $\widehat{\rho}_4$ to be decomposed is Hermitian. For this class of matrices (and for all normal matrices), the so-called geometric multiplicity of each eigenvalue is guaranteed to be equal to its so-called algebraic multiplicity, i.e. these matrices are nondefective (see details in [50] p. 57, 58, 103). More explicitly, this means that each eigenvalue of $\widehat{\rho}_4$ is here known to appear d_1 times at arbitrary locations on the diagonal of D_4 (algebraic multiplicity), since these eigenvalues are the same, in a different order, as in another eigendecomposition of the same matrix $\widehat{\rho}_4$ (neglecting estimation errors), namely the eigendecomposition (1). Therefore, the dimension of the eigenspace associated with any eigenvalue of $\widehat{\rho}_4$ is also equal to d_1. So, when performing the eigendecomposition of $\widehat{\rho}_4$, the considered decomposition algorithm should provide d_1 linearly independent eigenvectors for each eigenvalue, and we hereafter assume that practical eigendecomposition algorithms succeed in finding them (that was indeed the case in all the tests reported further in this paper). As an overall result, when performing the eigendecomposition of $\widehat{\rho}_4$, 1) we get a diagonal matrix D_4 with $d = d_1 d_2$ eigenvalues that may be in an arbitrary order, and with each encountered value repeated d_1 times and 2) we get a unitary matrix V_4 whose columns are eigenvectors that have the above-defined properties.

As in Sect. 2.1, we then apply a joint permutation to the diagonal elements of D_4 and to the columns of V_4, so that the values corresponding to the diagonal elements of D_4 become in nonincreasing order. The matrix thus derived from V_4, with reordered eigenvectors in its columns, is here again denoted as $\widehat{U}_2$. The associated properties are detailed in Appendix F. This shows that this first part of our second type of methods does not yet succeed in separately identifying each column of U, but it succeeds in finding a basis for each of the d_2 subspaces $\mathcal{S}_{1,m_1}$ that corresponds to a value m_1, with $m_1 \in \{1, \ldots, d_2\}$, and that is associated with the d_1 columns of U that have the indices defined by (F8). We hereafter show how to extend that method so as to identify each column of U separately.

3.3 Second part of the methods

The two important properties that we showed in Sect. 3.2 for the considered (possibly high) dimension $d = d_1 d_2$ are as follows. First, we showed that we can "partition" the set of d columns of U in d_2 subsets, in the sense that , in Sect. 3.2, we identified each d_1-dimensional subspace $\mathcal{S}_{1,m_1}$ spanned by the columns of U that compose the subset with index m_1. Second, we showed that we can select which partition we perform: the columns of U that are gathered in the same subset are those that correspond to the same eigenvalue of ρ_2 and hence $\widehat{\rho}_4$.

In the present section, we take advantage of these results to create another partition of the columns of U, in order to then jointly exploit these two partitions as explained further in Sect. 3.4. We thus create a two-stage algorithm in the sense that it successively uses two eigendecompositions of density matrices estimated at the output of the process U, namely the eigendecompositions respectively described in Sect. 3.2 and in the present subsection. This should be contrasted with the algorithms of Sect. 2, that are single-stage methods in the sense that they perform a single eigendecomposition.

More precisely, the input density matrix ρ_5 here applied to the considered unitary process is different from ρ_1 but, here again, it is diagonal, it contains d_2 different values that each appear d_1 times and these values are the same as in Sect. 3.2. However, they are here placed in a different order: the set of values on the main diagonal of ρ_5 consists of a series of d_1 identical subsets, each of which successively contains the d_2 values $r_1, \ldots, r_{d_2}$. The density matrix ρ_5 thus reads

$$\rho_5 = \begin{bmatrix} r_1 & 0 & \ldots & & & & 0 \\ & \ddots & & & & & \\ & & r_{d_2} & & & & \\ & & & r_1 & & & \\ & & & & \ddots & & \\ & & & & & r_{d_2} & \\ 0 & & & & & & \ddots \end{bmatrix}. \tag{20}$$

This may also be expressed as

$$\rho_5 = I_{d_1} \otimes \mathrm{diag}(r) \tag{21}$$

with the same notations as in (18).

This input density matrix ρ_5 is used in the same way as in Sect. 3.2. So, the density matrix ρ_6 of the corresponding process output may be derived from ρ_5 in the same way as in (1). An estimate $\widehat{\rho}_6$ of ρ_6 is derived by a QST algorithm and, if required by the considered QST algorithm, we preprocess $\widehat{\rho}_6$ in order to derive: 1) its Hermitian version $\widehat{\rho}_7$ (in the same way as in (B5)) and then 2) its Hermitian and unit-trace version $\widehat{\rho}_8$ (in the same way as in (B6)). We then perform an eigendecomposition of $\widehat{\rho}_8$. This yields a diagonal matrix D_8 containing its eigenvalues and a matrix V_8 whose columns are its associated eigenvectors.

We then apply a joint permutation to the diagonal elements of D_8 and to the columns of V_8, so that the values corresponding to the diagonal elements of D_8 become in nonincreasing order. Here again, each of the d_2 successive subsets of d_1 identical values along the diagonal of the reordered version of D_8 (values equal to eigenvalue r_{m_2}) has an index m_2, with $m_2 \in \{1, \ldots, d_2\}$, and consists of elements with indices

$$(m_2 - 1)d_1 + n_2, \quad \text{with} \quad n_2 \in \{1, \ldots, d_1\}. \tag{22}$$

Moreover, the corresponding matrix, derived from V_8, with reordered eigenvectors in its columns is denoted as $\widehat{U}_3$

[3] Its subset of columns that also have the indices defined by (22) is a set of d_1 eigenvectors that form a basis of the d_1-dimensional eigenspace of $\widehat{\rho}_8$ associated with its eigenvalue r_{m_2}. This eigenspace, obtained in the second part of this method, is hereafter denoted as $\mathcal{S}_{2,m_2}$.

$\mathcal{S}_{2,m_2}$ also has a relationship with the actual matrix U to be identified, but it is different from the relationship disclosed for $\mathcal{S}_{1,m_1}$ in Sect. 3.2. Here again, $\mathcal{S}_{2,m_2}$ is the d_1-dimensional subspace spanned by the d_1 columns of U that correspond to the eigenvalue r_{m_2}. However, here the indices of these columns are different from those of Sect. 3.2: they are equal to the column indices of all occurrences of r_{m_2} in (20), that is

$$m_2 + n_2 d_2 \quad \text{with} \quad n_2 \in \{0 \ldots, d_1 - 1\}. \tag{23}$$

This shows that this second part of our second type of methods succeeds in finding a basis for each of the d_2 subspaces $\mathcal{S}_{2,m_2}$ that: 1) corresponds to a value m_2, with $m_2 \in \{1, \ldots, d_2\}$, and 2) is associated with the d_1 columns of U that have the indices defined by (23). We hereafter show how to combine the properties of these subspaces $\mathcal{S}_{2,m_2}$ and of the subspaces $\mathcal{S}_{1,m_1}$ derived in the first part of this algorithm.

3.4 Third part of the methods

We now consider all subspace intersections $\mathcal{S}_{1,m_1} \cap \mathcal{S}_{2,m_2}$, for $m_1 \in \{1, \ldots, d_2\}$ and $m_2 \in \{1, \ldots, d_2\}$. The subspaces $\mathcal{S}_{1,m_1}$ and $\mathcal{S}_{2,m_2}$ are each defined by a set of column vectors of U and we want each such pair of subspaces to share *only one* column vector (or possibly no columns for some of these intersections). When this condition is met, $\mathcal{S}_{1,m_1} \cap \mathcal{S}_{2,m_2}$ is equal to the one-dimensional subspace defined by this shared column vector, because U is unitary and all its column vectors are therefore orthogonal (see [50] p. 67). Moreover, we want the set of d_2^2 intersection column vectors thus created to yield all d columns of U (up to scale factors). This requires $d_2^2 \geq d$ and hence, due to (16):

$$d_1 \leq d_2. \tag{24}$$

Besides, the number d_2 of different eigenvalues of the input density matrix (as defined in Sect. 3.2) is upper bounded by the value n_{max}, as explained above. Therefore, due to (24 and 16), the state space dimension d that can be handled by the methods proposed in the present section is upper bounded by:

$$d \leq (n_{max})^2. \tag{25}$$

This maximum space dimension is therefore much higher than the bound (15) faced with the methods of Sect. 2, under the assumption that n_{max} here remains the same

[3] all the quantities of importance related to estimates of the considered process are denoted with the same letter and increasing indices 1, 2 and so on. When describing our first type of methods in Sect. 2, for the sake of clarity we used detailed notations for these estimates, therefore with a first notation (with index 1) for the quantity before column reordering and a second notation (with index 2) for the quantity after column reordering. However, we then did not use the first notation. We here want to make it clear that, for the second type of methods described in the present section, we also consider two such quantities, namely before and after reordering. However, we do not introduce a notation for the quantity before reordering because, here again, we would not use it afterwards. Therefore, we here only introduce a single quantity, with an index equal to 3.

as with single-stage methods, as discussed above. In that case, the methods of the present section are much more attractive for handling high-dimensional spaces (but more complex).

In the remainder of this section, we only consider the case when

$$d_1 = d_2 \tag{26}$$

which makes the analysis simpler. Combining this condition with (16) yields

$$d = d_1^2 = d_2^2. \tag{27}$$

When d_2 is moreover set to its maximum value n_{max}, we obtain $d = (n_{max})^2$, so that we reach the upper bound defined by (25).

One then analyzes how the indices of the columns of U corresponding to $\mathcal{S}_{1,m_1}$ (defined by (F8)) are interleaved with the indices of the columns of U corresponding to $\mathcal{S}_{2,m_2}$ (defined by (23)), for any given couple of values (m_1, m_2), with $m_1 \in \{1, \ldots, d_2\}$ and $m_2 \in \{1, \ldots, d_2\}$. This shows that, whatever (m_1, m_2), the sets of columns of U respectively associated with $\mathcal{S}_{1,m_1}$ and $\mathcal{S}_{2,m_2}$ share one and only one column, and that its index is

$$(m_1 - 1)d_1 + m_2. \tag{28}$$

Moreover, when m_1 and m_2 are varied over all their ranges $m_1 \in \{1, \ldots, d_2\}$ and $m_2 \in \{1, \ldots, d_2\}$, the resulting set of d_2^2 indices (28) spans all $d = d_1 d_2 = d_2^2$ columns of U. This approach therefore succeeds in identifying all columns of U, up to scale factors at this stage.

The practical algorithm proposed for implementing the above approach operates as follows. Successively for each couple (m_1, m_2), with $m_1 \in \{1, \ldots, d_2\}$ and $m_2 \in \{1, \ldots, d_2\}$, we need to determine a vector situated in the 1-dimensional intersection of the subspaces $\mathcal{S}_{1,m_1}$ and $\mathcal{S}_{2,m_2}$. The data available in practice for defining $\mathcal{S}_{1,m_1}$ consist of the set of columns of $\widehat{U}_2$ that have the indices defined by (F8) and the data that define $\mathcal{S}_{2,m_2}$ consist of the set of columns of $\widehat{U}_3$ that have the indices defined by (22). To determine this 1-dimensional subspace intersection, we here use the first direction provided by the standard Canonical Correlation Analysis, or CCA, e.g. described in [56] and directly available in the Matlab scientific software [57]. This approach is by the way closely related to the eigendecomposition tool, that is the core of the methods proposed in the present paper.[4]

By calculating the intersection $\mathcal{S}_{1,m_1} \cap \mathcal{S}_{2,m_2}$, we obtain one column vector of U, possibly up to an arbitrary complex factor, unless this vector is guaranteed to be normalized by the considered CCA tool. If it is not, we then reduce this factor to a phase factor, by dividing this intersection vector by its norm, so as to enforce its norm to become equal to one (again because U is known to be unitary and thus to have unit-norm column vectors).

[4] It should also be noted that an alternative to CCA, based on Singular Value Decomposition, and that allows one to determine subspace intersections, is available in [58]. It is a building block of the method called SIBIS, for "Subspace-Intersection Blind Identification and Separation". Beyond its subspace-intersection tool, SIBIS has the following relationships with the present paper in terms of final goal. SIBIS performs 1) Blind Mixture Identification, or BMI [59–62], which is a type of classical system identification problem and is thus related to (the classical counterpart of) QPT and 2) Blind Source Separation, which is a problem closely connected to classical BMI.

All these normalized results of CCA are used to create an additional matrix related to the estimation of U and therefore denoted as $\widehat{U}_4$: each application of CCA to a given couple (m_1, m_2) yields a normalized vector that is stored in the column of $\widehat{U}_4$ whose index is defined by (28). The matrix $\widehat{U}_4$ obtained at this stage is therefore equal to U up to an arbitrary phase factor in each of its columns (and up to estimation errors).

3.5 Fourth part of the methods

The properties of $\widehat{U}_4$ defined at the end of the previous section are exactly the same as those obtained in our single-stage methods for $\widehat{U}_2$: see Sect. 2.1, including (4). Therefore, the final stage of the method considered here consists of processing $\widehat{U}_4$ in the same way as we processed $\widehat{U}_2$ in Sect. 2.2. This yields a matrix $\widehat{U}_5$ that here again succeeds in restoring U, up to only a global phase factor.

A pseudo-code of the version of our algorithm described in this Sect. 3 is provided in Algorithm 2. This algorithm is called EQPT2. Two variants (called EQPT3 and EQPT4) of two-stage QPT methods are moreover presented in Appendix G.

Input : a) Estimate $\widehat{\rho}_4$ of output density matrix ρ_2 (provided by QST and obtained for input density matrix (17) with the diagonal values defined in (19)). b) Estimate $\widehat{\rho}_8$ of output density matrix ρ_6 (provided by QST and obtained for input density matrix (20) with the diagonal values defined in (19)). c) Estimate $|\widehat{\Psi}_2\rangle$ of output ket $|\Psi_2\rangle$ (provided by QST and obtained for all components of input ket $|\Psi_1\rangle$ equal to $1/\sqrt{d}$).

Output: Estimate $\widehat{U}_5$ of quantum process matrix U.

begin

`/* Exploit` $\widehat{\rho}_4$`: */`

Eigendecomposition: derive 1) a diagonal matrix D_4 that contains the eigenvalues of $\widehat{\rho}_4$ in an arbitrary order and 2) a matrix V_4 whose columns are eigenvectors of $\widehat{\rho}_4$ in the same order as eigenvalues;

Reorder the eigenvalues in D_4 in nonincreasing order and apply the same permutation to the columns of V_4 to create the matrix $\widehat{U}_2$;

`/* Exploit` $\widehat{\rho}_8$`: */`

Eigendecomposition: derive 1) a diagonal matrix D_8 that contains the eigenvalues of $\widehat{\rho}_8$ in an arbitrary order and 2) a matrix V_8 whose columns are eigenvectors of $\widehat{\rho}_8$ in the same order as eigenvalues;

Reorder the eigenvalues in D_8 in nonincreasing order and apply the same permutation to the columns of V_8 to create the matrix $\widehat{U}_3$;

`/* Combine` $\widehat{U}_2$ `and` $\widehat{U}_3$`: */`

for $m_1 = 1$ *to* d_2 **do**

  for $m_2 = 1$ *to* d_2 **do**

    Set column of $\widehat{U}_4$ with index equal to $((m_1 - 1)d_1 + m_2)$: Set it to first vector provided by CCA of 1) the set of columns of $\widehat{U}_2$ with indices $((m_1 - 1)d_1 + n_1)$ with $n_1 \in \{1, \ldots, d_1\}$ and 2) the set of columns of $\widehat{U}_3$ with indices $((m_2 - 1)d_1 + n_2)$ with $n_2 \in \{1, \ldots, d_1\}$;

    `/* If the CCA tool does not provide unit-norm vectors, then do it: */`

    Divide above column of $\widehat{U}_4$ by its norm;

  end

end

`/* Exploit` $|\widehat{\Psi}_2\rangle$`: */`

$|\Psi_3\rangle = \widehat{U}_4^{\dagger}|\widehat{\Psi}_2\rangle$;

$\widehat{U}_5 = \widehat{U}_4 \operatorname{diag}(|\Psi_3\rangle \oslash |\Psi_1\rangle)$;

end

Algorithm 2 EQPT2: second Eigenanalysis-based QPT algorithm (composed of two stages). This version is for the case when the QST algorithm provides estimates that meet the known properties of the actual quantities. Otherwise, see text.

4 Proposed multi-stage QPT methods

The QPT methods proposed in the previous sections can be extended as follows, by selecting density matrices ρ_1 of input mixed states that are diagonal , that contain *several occurrences* of each value that appears on their diagonal and that have additional properties. More precisely, here again, each value that appears on the main diagonal of ρ_1 appears d_1 times , with $d_1 > 1$, and that diagonal contains d_2 different values, here possibly with $d_2 \neq d_1$. The number of rows of ρ_1, equal to the dimension of the considered state space, is thus here again defined by (16). The new feature introduced here consists

of storing all d_1 occurrences of any given diagonal value of ρ_1 in b_n blocks of adjacent rows, with each block consisting of b_s rows, and hence with

$$b_n b_s = d_1. \tag{29}$$

Besides, we consider the case when b_n, b_s and d_1 are powers of 2. We denote as b_ℓ the integer-valued logarithm of b_n in base 2: we have

$$b_n = 2^{b_\ell}. \tag{30}$$

We hereafter consider all possible integer values of b_ℓ, for a fixed value d_1 and varying values of b_n and b_s. In other words, we consider all linked powers of 2 for b_n and b_s, in their acceptable range. The minimum of b_ℓ is 0, corresponding to the minimum of b_n, equal to 1, and to the maximum of b_s, equal to d_1. Similarly, the maximum of b_ℓ is equal to $\log_2 d_1$, corresponding to the maximum of b_n, that is equal to d_1, while b_s reaches its minimum, equal to 1.

Moreover, these blocks are organized as follows: from the top to the bottom of the diagonal of ρ_1, we have a block of values equal to r_1, then a block of values r_2, and so on until a block of values r_{d_2}, and then the same structure again and again, that is, blocks of values $r_1, r_2, \ldots, r_{d_2}$ (when there exist several blocks containing the same value, i.e. when $b_n > 1$). Here again, we request these possible values to be defined in decreasing order, i.e. $r_1 > r_2 > \cdots > r_{d_2}$, and to take the same values as in (19). The two extreme cases of this general framework are nothing but the two specific cases that we already used for the above-defined two-stage methods: when setting $b_\ell = 0$ and hence $b_n = 1$ and $b_s = d_1$, we obtain the density matrix of (17) whereas with $b_\ell = \log_2 d_1$ and hence $b_n = d_1$ and $b_s = 1$, we get the density matrix of (20).

The main idea of the above-defined two-stage methods was to exploit intersections of two subspaces of eigendecompositions resulting from the above density matrices (17 and 20). Here, we extend this idea to general multi-stage methods, with $(\log_2 d_1 + 1)$ stages. To this end, we consider all $(\log_2 d_1 + 1)$ input density matrices that belong to the above-defined general framework, by considering all $(\log_2 d_1 + 1)$ possible values of b_ℓ, i.e. all integers from 0 to $\log_2 d_1$. Using the same approach as in Sect. 3 (that is, performing the eigendecomposition and reordering for the estimated output density matrix), for each value of b_ℓ and hence of b_n and b_s, we obtain a set of d_2 subspaces denoted as $\mathcal{S}_{b_\ell, m_{b_\ell}}$, with $m_{b_\ell} \in \{1, \ldots, d_2\}$ (each index $m_{b_\ell}$is used to define each subspace in the eigendecomposition of the stage $b_\ell$of the algorithm proposed here; its correspondence with the structure of the input density matrix is not the same as when we used the indices m_1 and m_2 in Sect. 3). These subspaces $\mathcal{S}_{b_\ell, m_{b_\ell}}$ are defined in more detail in Appendix H. We then extend the approach of Sect. 3 by performing subspace intersections as follows. We successively consider all $d_2^{(\log_2 d_1 + 1)}$ possible values of the set of $(\log_2 d_1 + 1)$ subspace indices m_{b_ℓ} (with $m_{b_\ell} \in \{1, \ldots, d_2\}$), i.e. we consider all possible values of the set of indices $\{m_0, \ldots, m_{\log_2 d_1}\}$. For each such set of indices, we derive the intersection of the corresponding $(\log_2 d_1 + 1)$ subspaces $\mathcal{S}_{b_\ell, m_{b_\ell}}$. We thus form $d_2^{(\log_2 d_1 + 1)}$ subspace intersections. As in Sect. 3.4, we want each of these intersections to be one-dimensional (or possibly to be restricted to the null vector for some of these intersections). In order to identify all d columns of U, the number of subspace intersections should be at least equal to d. Using (16), this condition reads

$$d_2^{\log_2 d_1} d_2 \geq d_1 d_2. \tag{31}$$

Taking the logarithm, in base 2, of that equation yields

$$\log_2 d_2 \geq 1 \tag{32}$$

that is

$$d_2 \geq 2. \tag{33}$$

In particular, for $d_2 = 2$, we have equalities in (33) back to (31) and the number $d_2^{(\log_2 d_1 + 1)}$ of intersections is exactly equal to the dimension (16). It should be noted that, unlike in the two-stage methods, we here do not face the constraints (24) and hence (25). This is because we pay the price by using a multi-stage, hence more complex, approach: the method proposed here potentially applies to any state space dimension d and any number d_2 of different eigenvalues (with (33)), provided we accept to set accordingly the value of d_1, defined by (16), and hence the number $(\log_2 d_1 + 1)$ of stages of the method and of eigendecompositions to be performed (and, again, provided this method remains numerically accurate enough, despite its higher computational complexity).

Moreover, also in this multi-stage extension, we aim at reducing the number d_2 of different eigenvalues, because we thus hope to better succeed in assigning each estimated eigenvalue to the correct value among the possible ones, despite estimation errors. A very attractive feature of the approach proposed here it that it allows one to decrease d_2 to 2, thus leading to Dichotomic multi-stage Eigenanalysis-based QPT, or DEQPT, methods (where "dichotomic" refers to the fact that the complete set of eigenvalues is split into two subsets, with identical values in each subset).

In Appendix H.1 we show that, for $d_2 = 2$, the above-defined subspace intersections are each indeed restricted to one dimension and yield all columns of U (up to scale or phase factors). Moreover, we illustrate this property and thus the proposed complete multi-stage methods in Appendix H.2. Finally, Appendix H.3 contains a pseudo-code for that multi-stage QPT method applicable to an arbitrary state space dimension and called EQPT5.

5 Test results

To validate the EQPT1, EQPT2, EQPT3 and EQPT5 methods detailed in the previous sections, and to evaluate their accuracy, we performed numerical tests with data derived from a software simulation of the considered configuration. As mentioned in Sect. 1, all the methods proposed in the present paper belong to the class of QPT methods that first perform a complete QST for each considered process output state and then exploit the results of these QST to achieve the core of QPT. The input of our QPT methods thus consists of estimates (more precisely, models of such estimates, as detailed below) of pure or mixed states provided by QST. Various QST methods were reported in the literature (see e.g. [1, 9, 11, 63–77]) and one may hereafter e.g. use any of the available QST methods compatible with the types of quantum states encountered in each of the QPT methods proposed in the present paper. Therefore, to here provide numerical results that define the accuracy of our QPT methods, we do not restrict this investigation to the use of a single QST method, from which the reader could not easily extrapolate to his preferred QST approach, with his preferred parameter values (especially, the number

of times he prepares a copy of a considered input state and he performs a measurement for the corresponding output state), whereas these parameter values may have a strong influence on QPT performance. Instead, we create a numerical model of quantum state estimates with a given accuracy and we focus on the resulting accuracy of our QPT methods, not only to estimate these absolute accuracies, but mainly to determine which of these methods yields the best performance when they use the *same* accuracy for QST. This moreover allows us to investigate the numerical performance of our QPT methods over a wide range of conditions, by varying the accuracy of the modeled quantum state estimates.

It should also be noted that the complexity of QST depends on the considered type of quantum state and on the selected QST algorithm. The most standard QST algorithm is detailed in Section 8.4.2 of [1]. It applies to the most general type of quantum states, namely mixed states, and requires computing the averages of $4^q - 1$ different types of measurements, where q is the number of qubits of the considered state [76]. Still for mixed states, a lower number of types of measurements can be reached, e.g. by considering states whose density matrix has a low rank: see [76] and references therein; see also [77] for the use of reduced density matrices. Moreover, various approaches only intended for the more specific case of pure states have been proposed and yield much lower numbers of types of measurements than the above standard approach for mixed states: see e.g. the recent overview of that topic and associated references in [76]. More specifically, some papers from the literature focus on reducing the complexity of QST algorithms (see e.g. [74, 75]). The overall complexity of a QST-based QPT algorithm therefore does not necessarily come from its QST part, as opposed to its subsequent part that consists of using QST results for the core of QPT. This also motivates us to perform tests as explained above, in order to uncouple the investigation of the complexity of the core of QPT (e.g. expressed in terms of required CPU time) from that of QST.

More precisely, to model the estimation errors that an actual QST method would yield when deriving a ket estimate (e.g. for the ket $|\Psi_2\rangle$ of Sect. 2.2), we first compute the theoretical value of that ket in the considered conditions (e.g. using (6) with the input ket defined before (12) and a known process matrix U) and we then add random, independent, complex-valued fluctuations to all components of this ket. Their real and imaginary parts are separately created with a random generator that has a probability density function that is uniform over the range $[-w/2, w/2]$, where the parameter w defines the magnitude of the modeled estimation fluctuations . This simple error model has the advantage of being easily interpretable. The parameter w is hereafter varied. This first allows us to investigate the sensitivity of the proposed QPT methods to the magnitude of QST estimation errors. Moreover, this allows us to make, as follows, a simple connection with actual QST methods. Due to the above-defined distribution of modeled fluctuations, the variance of their real and imaginary parts is equal to $w^2/12$. This value may e.g. be used to compare the results provided in Fig. 1 to what may be expected from the use a given QST method, by selecting the plots of Fig. 1 obtained for a value of w that yields almost the same variance of estimation errors as the considered QST method. The error model for density matrix estimation is defined similarly and is presented in Sect. 7 of the Supplementary Document. We stress that our QPT methods are operated without having any knowledge of the type of "noise" (i.e. estimation errors) that is faced, they were not designed for a specific type of noise and they are therefore versatile.

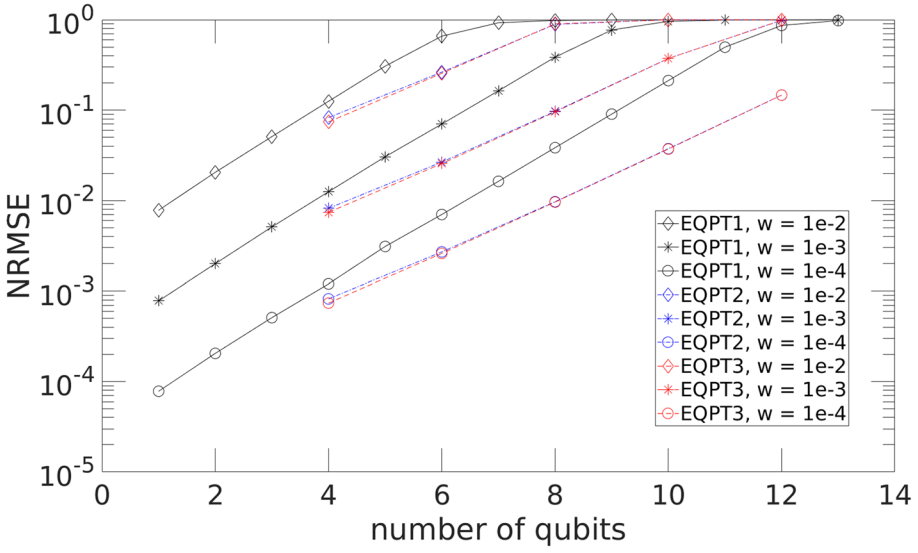

Fig. 1 Normalized Root Mean Square Error (NRMSE) of estimation of process matrix U of proposed Quantum Process Tomography (QPT) methods EQPT1 (black solid lines), EQPT2 (blue dash doted lines) and EQPT3 (red dashed lines), vs. number q of qubits, for several values of the parameter w that defines the magnitude of the estimation errors of Quantum State Tomography (QST)

Each elementary test uses a randomly drawn $d \times d$ process matrix U, obtained as follows. We first create a $d \times d$ real-valued matrix with each element randomly, uniformly, drawn in the interval [0, 1]. We then perform a QR decomposition of this matrix (e.g. using one of the algorithms defined in [50] p. 112, [78] p. 223, [79]). This especially yields the so-called "Q factor" of that QR decomposition (see e.g. the software package [79]) . This factor is guaranteed to be a unitary matrix (see [50] p. 112 , [78] p. 223, [79]) and we therefore use it as our matrix U.

Each above-defined matrix U is used by feeding it with the process input states defined in the previous sections, computing the corresponding exact output states (e.g. using (1) and (6)), deriving their above-defined modeled estimates associated with QST and applying the considered QPT method to those estimates, so as to derive the process matrix estimate $\widehat{U}_5$.

The criterion used in the present section to evaluate the performance obtained in such a test is defined as follows (and other criteria are studied in Appendix I). We aim at measuring how close we get to the case when $\widehat{U}_5$ is equal to U up to a phase factor. The criterion used to this end is defined by first introducing

$$NMSE(U, \widehat{U}_5) = \min_{\theta} \left[\frac{1}{2d} ||U - e^{i\theta} \widehat{U}_5||_F^2 \right] \tag{34}$$

where $||.||_F$ stands for the Frobenius norm and where the phase θ is chosen so as to minimize this NMSE. At first sight, it may not be obvious that one can derive *analytical* expressions of 1) the optimum value of the angle θ involved in (34) and 2) the corresponding value of (34). Fortunately, these expressions can be derived, as follows. Denoting the matrix trace as $\mathrm{Tr}(.)$ and using the property (see [80] p. 156)

$$||M||_F^2 = \mathrm{Tr}(M^\dagger M) \tag{35}$$

for any matrix M makes it possible to transform the expression $||U - e^{i\theta}\widehat{U}_5||_F^2$ in (34) and to show that it is minimized when θ takes the value

$$\theta_{optfro} = -\arg(\mathrm{Tr}(U^\dagger \widehat{U}_5)) = \arg(\mathrm{Tr}(\widehat{U}_5^\dagger U)). \tag{36}$$

The resulting value of the NMSE in (34) may be shown to be equal to

$$NMSE(U, \widehat{U}_5) = \frac{1}{2d}\left[||U||_F^2 + ||\widehat{U}_5||_F^2 - 2\,|\mathrm{Tr}(U^\dagger \widehat{U}_5)|\right]. \tag{37}$$

This criterion is a Normalized Mean squared Error (NMSE) since it is based on the sum of the squared moduli of the elements of the error matrix $(U - e^{i\theta}\widehat{U}_5)$, then normalized by a scale factor chosen so that the maximum value of this parameter is equal to one when U and $\widehat{U}_5$ are unitary (because their squared norms are then equal to d). The criterion that we eventually consider is the Normalized Root Mean Square Error (NRMSE) associated with the above NMSE:

$$NRMSE(U, \widehat{U}_5) = \sqrt{NMSE(U, \widehat{U}_5)}. \tag{38}$$

For each dimension d of the state space, the results reported below were obtained by running 100 above-defined elementary tests, each with a different, randomly drawn, process matrix U (except that we only performed 10 tests when applying the EQPT5 method to 11 qubits, as explained below). The performance figure then provided is the mean of $NRMSE(U, \widehat{U}_5)$ over all these tests. This protocol is repeated for various values of the number of qubits and hence of d.

The reported tests were carried out with a very basic, 8-year old, PC (Intel(R) Core(TM) i5-7500 CPU running at 3.40 GHz with 16 GB of RAM), using the Matlab software environment. The results reported below therefore show that the complexity of the proposed methods (computation time and memory size) remain acceptable over a large number of configurations: these results were obtained without having to resort to, e.g., High-Performance Computing (HPC) or at least Graphical Processing Units (GPUs).

As a first step, Fig. 1 shows the results thus obtained with our single-stage and two-stage methods, namely EQPT1 (black solid lines), EQPT2 (blue dash doted lines) and EQPT3 (red dashed lines). The results for EQPT1 are provided for all values of the state space dimension d that take the form $d = 2^q$, where q is the equivalent number of qubits, that is here varied from 2 to 13. $q = 13$ is the maximum number that was accepted in the considered hardware and software environment without running out of memory (for non-optimized code). The same approach applies to EQPT2 and EQPT3 except that, due to (27), only even values of q were considered. Moreover, for EQPT2 and EQPT3, performance turned out to often be somewhat degraded for the lowest possible state space dimension, namely $d = 4$, that corresponds to $q = 2$ qubits. This is not an issue, since EQPT2 and EQPT3 were developed to address higher values of d (EQPT1 can be used instead for lower values of d). Therefore, for better readability, the results of EQPT2 and EQPT3 for $q = 2$ are skipped in Fig. 1. It should also be noted that the complexity of the configurations that could be tested was limited by memory requirements, as stated above, hence not by the computational load of the proposed algorithms, that remained reasonable up to $q = 13$. For instance, for each of the 100 process matrix estimations,

the average execution time used for a) creating data (thus only "modeling" QST instead of running a QST algorithm: see above), b) running the QPT algorithm and c) displaying results, was as follows, respectively for EQPT1, EQPT2 and EQPT3: 1) $\simeq 1$ ms, 2.6 ms and 2.8 ms for $q = 4$ qubits, 2) 22 ms, 170 ms and 200 ms for $q = 8$ qubits and 3) 23 s, 180 s and 230 s for $q = 12$ qubits. The execution times for the other numbers of qubits are provided in Table 1.

The following conlusions can be drawn from Fig. 1, when comparing the proposed methods for any given QST quality, defined by the considered value of w. The two-stage methods EQPT2 and EQPT3 yield almost the same performance, with a slight advantage for EQPT3 (as expected from Appendix G) , that is more significant for lower values of the space dimension d. Moreover, these two-stage methods always yield better and often much better performance than the single-stage method EQPT1, as expected from the beginning of Sect. 3. Up to a certain space dimension d, the gap between these two types of methods increases with the space dimension and hence with the associated number q of qubits: whereas all methods yield linear variations of the logarithm of NRMSE with respect to q (i.e. logarithm of d), the slope of these variations is significantly lower for EQPT2 and EQPT3, and these methods already start with a lower NRMSE than EQPT1 for the lowest values of d. Depending on the considered conditions, the EQPT2 and EQPT3 methods decrease the NRMSE by a factor up to 6, and hence the NMSE by a factor up to 36, as compared with EQPT1. Moreover, for the lowest QST error magnitude considered in these tests, for all analyzed state space dimensions, EQPT2 and EQPT3 yield an NRMSE limited to about 10^{-1} and hence an NMSE limited to about 10^{-2}.

We now move to the multi-stage method EQPT5, whose performance is shown in Fig. 2 (red dashed lines), together with the results already provided above for EQPT1 (black solid lines) and EQPT2 (blue dash doted lines). The results for EQPT5 are reported for values of the state space dimension d that take the form $d = 2^q$, where q is the equivalent number of qubits, varied as follows. For a large q, EQPT5 yields a significantly higher computational complexity than the single-stage and two-stage methods, due to the higher number of eigendecompositions it performs and to the associated program structure (recursive function with a binary tree structure, as explained in Appendix H). Unlike for the previous proposed methods, the applicability of EQPT5 to a large number of qubits was therefore more limited by its computational load than by its memory requirements, when using the considered basic hardware and software means. Therefore, to keep a reasonable simulation duration (for *all* estimations of the process matrix), EQPT5 was only tested up to $q = 10$ qubits when performing 100 above-defined elementary tests, and also for $q = 11$ qubits but then with only 10 elementary tests. Moreover, as with EQPT2, we hereafter skip the results for a low dimension, that is for $d \leq 4$, which is not the case of interest for a multi-stage method. In the considered range of space dimensions, the above-defined average execution time per process matrix estimation was e.g. as follows for EQPT5: 5.4 ms for $q = 4$ qubits, 4.4 s for $q = 8$ qubits, 490 s $\simeq$ 8 mn for $q = 10$ qubits and 7600 s $\simeq$ 2 h 07 mn for $q = 11$ qubits. The execution times for the other numbers of qubits are provided in Table 1.

Figure 2 yields the following conclusions for EQPT5. For the lowest considered values of the space dimension, EQPT5 starts with a somewhat higher NRMSE than EQPT1 and EQPT2. But, an attractive feature of EQPT5 is that its NRMSE then increases more slowly that those of the other methods, with respect to the space dimension. This again

Table 1 Execution times (averaged over tests with all values of w) of the EQPT1, EQPT2, EQPT3 and EQPT5 methods vs. number of qubits

Method	Number of qubits												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
EQPT1	$\simeq 1$ ms	$\simeq 1$ ms	$\simeq 1$ ms	$\simeq 1$ ms	2.2 ms	3.9 ms	7.2 ms	22 ms	68 ms	400 ms	3.1 s	23 s	170 s
EQPT2				2.6 ms		15 ms		170 ms		4.3 s		180 s	
EQPT3				2.8 ms		18 ms		200 ms		5.0 s		230 s	
EQPT5			2.3 ms	5.4 ms	27 ms	100 ms	580 ms	4.4 s	38 s	490 s	7600 s		

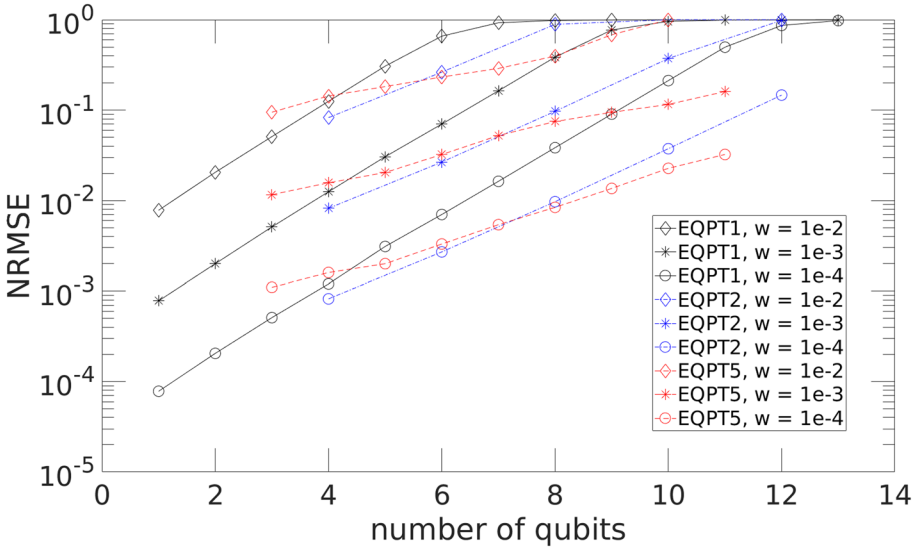

Fig. 2 Normalized Root Mean Square Error (NRMSE) of estimation of process matrix U of proposed Quantum Process Tomography (QPT) methods EQPT1 (black solid lines) , EQPT2 (blue dash doted lines) and EQPT5 (red dashed lines) , vs. number q of qubits for several values of the parameter w that defines the magnitude of the estimation errors of Quantum State Tomography (QST)

yields *linear* variations of the logarithm of NRMSE with respect to the number of qubits. EQPT5 thus outperforms the other two methods for higher space dimensions, and the gap between them increases with that dimension, up to a certain dimension. EQPT5 is thus attractive for high space dimensions, so that it actually reaches the goal for which it was designed. Depending on the considered conditions, EQPT5 decreases the NRMSE by a factor up to 3.2, and hence the NMSE by a factor up to more than 10, as compared with EQPT2. Moreover, for the lowest QST error magnitude considered in these tests, for all analyzed state space dimensions (which are slightly lower than those considered above for EQPT1 and EQPT2, however), EQPT5 yield an NRMSE limited to around 3.2×10^{-2} and hence an NMSE limited to around 10^{-3}.

6 Conclusion and future works

Quantum Process Tomography (QPT) is a research topic of major interest, due to its importance for experimentally characterizing the actual behavior of quantum gates. In the present paper, we addressed QPT configurations that have two features: first, we consider unitary processes, and second, we aim at developing QPT methods that are applicable to a high state space dimension, corresponding to a significant number of qubits, as explained in Sect. 1. The proposed QPT methods allow, but do not require, the considered process to be dense (i.e., non-sparse).

Since the considered process is unitary, its input–output relationship takes a specific form, that suggests that eigenanalysis may be used to estimate at least part of the parameters of the process matrix. We first built upon that idea, and we then extended it so as to also estimate the other process parameters that a plain eigenanalysis cannot estimate. This resulted in a first class of QPT methods (including EQPT1), that are stated to be single-stage since they only require one eigendecomposition.

The above approach may however be expected to yield limited performance when considering high-dimensional state spaces, due to the estimation errors of the Quantum

State Tomography (QST) that it uses as a building block. We very significantly reduced this problem by introducing multi-stage eigenanalysis-based extensions of our approach, where the term "multi-stage" refers to the fact that these methods successively perform several eigendecompositions. We explained in detail how two-stage methods (especially EQPT2 and EQPT3) operate, and then how to extend them to an arbitrary number of stages in order to minimize their sensitivity to QST errors (especially with the dichotomic EQPT5 method).

The relevance of this approach was validated by means of numerical tests performed with a simulated version of this QPT problem. This especially showed that, depending on the considered conditions, the tested two-stage methods (EQPT2 and EQPT3) decrease the NMSE by a factor up to 36, as compared with the tested single-stage method (EQPT1). Moreover, for the lowest QST error magnitude considered in these tests, for all analyzed state space dimensions, the two-stage methods yield an NMSE limited to about 10^{-2}. Besides, the multi-stage dichotomic EQPT5 method may yield a higher accuracy (NMSE decreased by a factor up to 10 as compared with the two-stage EQPT2 method and NMSE limited to about 10^{-3} in almost the same conditions as above) but at the expense of a higher computational load.

We here had to restrict ourselves to tests performed on a standard PC. Our future works will therefore especially include additional tests, performed on a much more powerful platform, to derive the performance of our methods for a significantly higher number of qubits. Anyway, the configurations reported in the present paper already involve a quite significant number of qubits (up to 11 to 13, depending on the considered method) and hence a high state space dimension d (up to a few thousands) as compared with various investigations from the literature, in particular with a qudit corresponding to only $d = 5$ in [3] that also deals with unitary processes. Our tests also showed that the applicability of our single-stage and two-stage methods was not limited by their complexity in terms of computational cost, unlike various alternative QPT methods, but only by the memory size available on the platform that was used for running them. Our dichotomic multi-stage methods have the opposite features.

Appendix A Mathematical definitions and properties

Definition 1 The element-wise division $\oslash$ for vectors is defined as follows. Considering two vectors u and v, respectively with k th elements u_k and v_k, the quantity $u \oslash v$ is the vector whose k th element is equal to u_k/v_k.

Theorem 1 Every normal matrix $A \in M_n$ can be written as

$$A = U\Lambda U^{\dagger}, \tag{A1}$$

where $U \in M_n$ is unitary and $\Lambda \in M_n$ is a diagonal matrix whose main diagonal entries are the eigenvalues of A (M_n is the set of n-by- n complex matrices).

Proof See [50] p. 157 (e). See also [50] p. 101 for unitarily diagonalizable matrices (every normal matrix is unitarily diagonalizable) and the spectral theorem for normal matrices. In particular, this implies the spectral theorem for Hermitian matrices: see [50] p. 104. □

Theorem 2 If $A \in M_n$ reads

$$A = PDP^{-1} \tag{A2}$$

where $D \in M_n$ is a diagonal matrix and $P \in M_n$ is invertible then, for any index $k \in \{1, \ldots, n\}$, the k th column of P is an eigenvector of A and the associated eigenvalue is the k th value on the diagonal of D.

Proof Right-multiplying (A2) by P yields

$$AP = PD. \tag{A3}$$

For any index $k \in \{1, \ldots, n\}$, the k th column of the matrix equation (A3) reads

$$Av_k = \lambda_k v_k \tag{A4}$$

where v_k is the k th column of P and λ_k is the k th element on the diagonal of D. Eq. (A4) shows that v_k is an eigenvector of A and that the associated eigenvalue is λ_k. □

Theorem 3 If $A \in M_n$ is normal, and if x and y are eigenvectors corresponding to distinct eigenvalues, x and y are orthogonal.

Proof See [50] p. 103. □

Theorem 4 A best least-squares approximation of a given matrix $A \in M_n$ by a unitary matrix $U_A \in M_n$ is given by $U_A = VW^\dagger$, where $A = V\Sigma W^\dagger$ is a singular value decomposition (SVD) of A.

Proof See [50] p. 431. □

Appendix B Preprocessing steps of the single-stage QPT algorithm

The first part of our single-stage algorithm, described in Sect. 2.1, is based on an estimated density matrix $\widehat{\rho}_2$ provided by a QST algorithm. If this estimate $\widehat{\rho}_2$ is not guaranteed to be Hermitian due to the estimation errors of the considered "external" (i.e. external to *our* QPT) QST algorithm, a first recommended preprocessing step of our method consists of deriving a Hermitian estimate $\widehat{\rho}_3$ from $\widehat{\rho}_2$. This may be achieved by computing the Hermitian part of $\widehat{\rho}_2$, that is (see [50] p. 109 or [52] p. 343 for its real-valued counterpart):

$$\widehat{\rho}_3 = \frac{1}{2}\left(\widehat{\rho}_2 + \widehat{\rho}_2^\dagger\right). \tag{B5}$$

Similarly, the actual matrix ρ_2 is known to have a trace equal to 1, since it is a density matrix (see e.g. [1] p. 101, [81] p. 73, [46] p. 283). Therefore, if its estimate ($\widehat{\rho}_2$ or) $\widehat{\rho}_3$ obtained so far is not guaranteed to have a unit trace due to QST estimation errors, a second recommended preprocessing step of our method consists of deriving a (still Hermitian and moreover) unit-trace estimate $\widehat{\rho}_4$ from ($\widehat{\rho}_2$ or) $\widehat{\rho}_3$, by dividing all its elements by its trace:

$$\widehat{\rho}_4 = \frac{\widehat{\rho}_3}{\mathrm{Trace}(\widehat{\rho}_3)}. \tag{B6}$$

Similarly, the second part of our single-stage algorithm, described in Sect. 2.2, uses an estimated ket $|\widehat{\Psi}_2\rangle$, provided by a QST algorithm. If the considered QST algorithm does not guarantee that the norm of $|\widehat{\Psi}_2\rangle$ is equal to 1, a recommended pre-processing step of our QPT method consists of dividing $|\widehat{\Psi}_2\rangle$ by its norm.

Appendix C Preferred input density matrix for the first part of the single-stage QPT algorithm

In the first step of our first method defined in Sect. 2.1, we preferably use the value of ρ_1 defined as follows. The values on its diagonal are the eigenvalues of the considered density operator ρ_2, they are real and non-negative and their sum is equal to one (see e.g. [1] p. 101). As shown in Sect. 2.1, the proposed method assumes that these eigenvalues are different and ordered, and it is based on reordering the eigenvalues in the eigendecomposition of the estimated output density matrix and on permuting the eigenvectors accordingly. Therefore, if using quite close actual eigenvalues when choosing ρ_1, one may fear that the corresponding estimated eigenvalues obtained for the output density matrix be permuted due to estimation errors, so that the corresponding eigenvectors may then be permuted accordingly. This is a major problem, because the intermediate estimate $\widehat{U}_2$ related to the considered process, obtained at the output of this first step of our QPT method, is primarily based on these eigenvectors. In this paper, we handle this problem as follows: first, for the method imposed in Section 2.1, we select the value of ρ_1 in order to limit that problem to the greatest extent that is possible within the frame of that method; then, in Sects. 3 and 4, we propose more advanced methods that allow one to further reduce that problem. So, for the method considered here, we choose eigenvalues that are uniformly distributed (i.e. multiples of the same step, with zero excluded) , so as to maximize the gap between any two adjacent values. Moreover constraining them to be positive, with a sum equal to one and in decreasing order as explained above, it is easily shown that one obtains the following values for the diagonal elements $\rho_{1,kk}$ of ρ_1:

$$\rho_{1,kk} = \frac{2(d-k+1)}{d(d+1)} \qquad k \in \{1,\dots,d\}. \tag{C7}$$

Appendix D Handling scale factors in the first part of the single-stage QPT algorithm

Once one has performed the processing steps defined in Sect. 2.1 before its reference to the present Appendix, the only indeterminacy that remains in $\widehat{U}_2$ with respect to U is an unknown complex scale factor in each of its columns, since any such column belongs to the adequate one-dimensional eigenspace and is therefore defined up to a complex scale factor. Moreover, we create $\widehat{U}_2$ so that each of its columns has unit norm. To this end, if the considered eigendecomposition algorithm does not directly guarantee that this unit-norm condition is met, we postprocess the value of $\widehat{U}_2$ that it yields, by dividing each column of $\widehat{U}_2$ by its norm. After this postprocessing, the scale factor of each column of $\widehat{U}_2$, with respect to the column of U that has the same index, reduces to a unit-modulus factor, i.e. a phase factor, because U is unitary and therefore all its columns have unit norm, as those of $\widehat{U}_2$. $\widehat{U}_2$ then reads as shown in (4).

Appendix E Preferred input ket for the second part of the single-stage QPT algorithm

In the second step of our first method defined in Sect. 2.2, we select the input ket $|\Psi_1\rangle$ as follows. Based on the condition on nonzero components provided in Sect. 2.2 for $|\Psi_1\rangle$, we preferably select a state $|\Psi_1\rangle$ with all component moduli as high as possible. Considering positive real-valued components and taking into account that $|\Psi_1\rangle$ has unit norm, this yields a single solution: all components of $|\Psi_1\rangle$ in the considered basis are equal to $1/\sqrt{d}$.

Appendix F Permutation and properties of the first part of the two-stage QPT algorithm

We here address the first part of the two-stage QPT method described in Sect. 3.2. After the processing steps described in that section, as in Sect. 2.1, we apply a joint permutation to the diagonal elements of D_4 and to the columns of V_4, so that the values corresponding to the diagonal elements of D_4 become in nonincreasing order. Considering the resulting complete set of values from the top to the bottom of the diagonal of the reordered version of D_4, we thus get d_2 successive subsets, with the same value repeated d_1 times in each subset (in practice, these d_1 values may be slightly different, due to estimation errors), and with different and decreasing values from one subset to the next one. In other words, up to estimation errors, the reordered version of D_4 is nothing but the matrix ρ_1 of (17). Each subset of identical values along the diagonal of the obtained matrix (values equal to eigenvalue r_{m_1}) has an index m_1, with $m_1 \in \{1, \ldots, d_2\}$, and consists of elements with indices[5]

$$(m_1 - 1)d_1 + n_1, \qquad \text{with} \qquad n_1 \in \{1, \ldots, d_1\}. \tag{F8}$$

Moreover, the corresponding matrix, derived from V_4, with reordered eigenvectors in its columns is here again denoted as $\widehat{U}_2$. Its subset of columns that also have the indices defined by (F8) is a set of d_1 eigenvectors that form a basis of the d_1-dimensional eigenspace of $\widehat{\rho}_4$ associated with its eigenvalue r_{m_1}. This eigenspace, obtained in the first part of this method, is hereafter denoted as $\mathcal{S}_{1,m_1}$.

Importantly, $\mathcal{S}_{1,m_1}$ also has the following relationship with the actual matrix U to be identified. Eigenvectors corresponding to distinct eigenvalues are orthogonal for a normal matrix (see Theorem 3) and hence especially for a Hermitian matrix (see also [50] p. 171). This has the following consequence, considering that (1) and (17) define an eigendecomposition of ρ_2: each eigenvalue r_{m_1} of ρ_2, with a multiplicity d_1, corresponds to a d_1-dimensional eigenspace 1) that is orthogonal to the eigenspaces defined by all other eigenvalues and 2) that is the subspace defined by the columns of U that have the indices defined by (F8). But this eigenspace is nothing but $\mathcal{S}_{1,m_1}$, since ρ_2 is equal to $\widehat{\rho}_4$ (up to estimation errors). This shows that this first part of our second type of methods does not yet succeed in separately identifying each column of U, but it succeeds in finding a basis for each of the d_2 subspaces $\mathcal{S}_{1,m_1}$ that corresponds to a value m_1, with $m_1 \in \{1, \ldots, d_2\}$, and that is associated with the d_1 columns of U that have the indices defined by (F8).

[5] In this whole paper except Appendix H, the index of the first row and column of a matrix is equal to 1.

Appendix G Variants of two-stage QPT methods

Since the columns of $\widehat{U}_4$ are estimated independently, one may fear that they are not exactly orthogonal, due to estimation errors. Then, $\widehat{U}_4$ and hence $\widehat{U}_5$ are not unitary, although they should be because they aim at estimating U, that is constrained to be unitary. A first solution to this problem consists of postprocessing $\widehat{U}_4$, so as to determine a best least-squares unitary approximation of $\widehat{U}_4$. An algorithm that may be used to this end is defined by Theorem 4. The resulting modified QPT method is called EQPT3 hereafter. A slightly different version consists of finding a unitary approximation of $\widehat{U}_5$ instead of $\widehat{U}_4$. The resulting QPT method is called EQPT4. The pseudo-codes of these two methods are provided in Sect. 6 of the Supplementary Document.

Appendix H Additional information for the proposed multi-stage methods

H.1 Relevance of the methods

In this appendix, we first show that the approach with $d_2 = 2$ that we proposed in Sect. 4, based on subspace intersections, is relevant.

Each subspace $\mathcal{S}_{b_\ell, m_{b_\ell}}$ defined in Sect. 4 is associated with a set of columns of the process matrix U to be identified, in the following sense. We consider a given arbitrary stage of our algorithm, defined by the index b_ℓ, and the subset of columns of U that correspond to (i.e. that have the same column indices as) a given arbitrary diagonal value $r_{m_{b_\ell}}$ of the input density matrix ρ_1 when forming (1) . This subset of columns of U and $r_{m_{b_\ell}}$ are defined by the index $m_{b_\ell} \in \{1, 2\}$. We then have the following property: the eigendecomposition plus reordering algorithm proposed in Sect. 4 yields a set of column vectors associated with b_ℓ and m_{b_ℓ}, that form a basis of a subspace denoted as $\mathcal{S}_{b_\ell, m_{b_\ell}}$, and that subspace is also the subspace spanned by the columns of U associated with the index m_{b_ℓ} in the above-defined sense.

The analysis of the indices of these columns presented in the present appendix is made easier by introducing the following modified notations, related to the expression of these indices in base 2, which is motivated by the structure of the matrix ρ_1 based on powers of 2 that was introduced in Sect. 4:

- The index of the first row and first column of ρ_1 and U is here equal to 0 (instead of 1 in the other parts of this paper).
- Each variable m_{b_ℓ}, used to split in two subsets the columns of ρ_1, takes the values 1 and 2 (because here $d_2 = 2$). With this, we here associate another binary variable, that takes the values 0 and 1, namely $\mu_{b_\ell} = m_{b_\ell} - 1$.
- When using the variable μ_{b_ℓ}, the subspaces are denoted with the letter $\mathcal{T}$, to avoid any misunderstanding with the notation $\mathcal{S}_{b_\ell, m_{b_\ell}}$ used above, especially when considering explicit numerical values of their indices: for instance, $\mathcal{T}_{0,0}$ is nothing but $\mathcal{S}_{0,1}$. More generally speaking, we have $\mathcal{T}_{b_\ell, \mu_{b_\ell}} = \mathcal{S}_{b_\ell, m_{b_\ell}}$, whatever the values of the indices.With these notations, using the description provided in Sect. 4, the indices of the columns of U corresponding to $\mathcal{T}_{b_\ell, \mu_{b_\ell}}$ may be shown to be

$$2b_s b_i + \mu_{b_\ell} b_s + \{0, \ldots, (b_s - 1)\} \quad \text{with} \quad b_i \in \{0, \ldots, (b_n - 1)\}. \tag{H9}$$

This is more easily interpreted by indexing the stages of the algorithm in the reverse order, thus introducing

$$b_r = b_{\ell m} - b_\ell \tag{H10}$$

where $b_{\ell m} = \log_2 d_1$ is the maximum value of b_ℓ and therefore

$$d_1 = 2^{b_{\ell m}}. \tag{H11}$$

Using (29, 30, H11 and H10), we obtain

$$b_s = 2^{b_r}. \tag{H12}$$

Also using (30), the column indices in (H9) may therefore be rewritten as

$$\begin{aligned} b_i 2^{(b_r+1)} + \mu_{b_\ell} 2^{b_r} + \{0, \ldots, (2^{b_r} - 1)\} \\ \text{with} \quad b_i \in \{0, \ldots, (2^{b_\ell} - 1)\}. \end{aligned} \tag{H13}$$

This should be compared to the representation in base 2 of the index of any column of U, that may be expressed as

$$\sum_{j=0}^{b_{\ell m}} c_j 2^j \tag{H14}$$

with each bit c_j equal to 0 or 1. Eq. (H14) allows one to interpret the set of indices in (H13), i.e. the set of indices of the columns of U that correspond to $\mathcal{S}_{b_\ell, m_{b_\ell}} = \mathcal{T}_{b_\ell, \mu_{b_\ell}}$, for given values of m_{b_ℓ}, b_ℓ and hence b_r, due to (H10): these column indices are all the indices whose representation in base 2 is such that their bit c_j with index $j = b_r$ is constrained to be equal to μ_{b_ℓ}, whereas all possible combinations are gathered in this set of index values, for all the bits c_j with indices $j \neq b_r$. To summarize, in $\mathcal{S}_{b_\ell, m_{b_\ell}}$, only the bit with index $j = b_r$ of the column index is imposed.

Then, we eventually aim at analyzing the intersections of $(b_{\ell m} + 1)$ subspaces introduced in Sect. 4. This may be rephrased as follows, using the notations introduced in the present appendix. Each considered intersection of $(b_{\ell m} + 1)$ subspaces $\mathcal{S}_{b_\ell, m_{b_\ell}} = \mathcal{T}_{b_\ell, \mu_{b_\ell}}$ corresponds to one value of the set of indices $\{m_0, \ldots, m_{b_{\ell m}}\}$ and therefore $\{\mu_0, \ldots, \mu_{b_{\ell m}}\}$. As explained above, separately for each value of b_ℓ and hence of b_r, the columns of U that are kept in this intersection are those whose index represented in base 2 has a bit with index $j = b_r$ that is fixed and equal to μ_{b_ℓ}. The intersection of the complete set of $(b_{\ell m} + 1)$ subspaces obtained when varying b_ℓ thus yields one and only one column of U, namely the one whose bits are all defined by the selected complete set of values $\{\mu_0, \ldots, \mu_{b_{\ell m}}\}$. This also entails that, when successively considering the subspace intersections corresponding to all values of $\{\mu_0, \ldots, \mu_{b_{\ell m}}\}$, all columns of U are successively obtained. This proves that the proposed method based on subspace intersections is guaranteed to completely identify U (here again up to one phase factor per column, these phase factors being then handled as in the other proposed methods).

The successive subspace intersections may be considered in such an order that the sets of indices $\{\mu_0, \ldots, \mu_{b_{\ell m}}\}$ correspond to the representation in base 2 of the integers in increasing order. Then, the columns of U are obtained with indices in increasing order.

This is exactly what is obtained in the successive rows of Table 2 in the example provided below.

H.2 Example

We here illustrate the above property and thus the proposed complete multi-stage methods by considering the case $d = 8$, $d_2 = 2$ and hence $d_1 = 4$ and $(\log_2 d_1 + 1) = 3$ stages. During the first stage, that corresponds to $b_\ell = 0$, the eigendecomposition allows one to identify two subspaces $\mathcal{S}_{0,m_0}$, respectively with $m_0 = 1$ and $m_0 = 2$, and that respectively correspond to the following sets of 4 indices of columns of U: $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$. Then, during the second stage, that corresponds to $b_\ell = 1$, the eigendecomposition allows one to identify two subspaces $\mathcal{S}_{1,m_1}$, respectively with $m_1 = 1$ and $m_1 = 2$, and that respectively correspond to the following sets of 4 indices of columns of U: $\{1, 2, 5, 6\}$ and $\{3, 4, 7, 8\}$. Finally, the third stage, that corresponds to $b_\ell = 2$, yields the following sets of 4 indices of columns of U: $\{1, 3, 5, 7\}$ and $\{2, 4, 6, 8\}$. We then derive the corresponding subspace intersections, first considering $\mathcal{S}_{0,m_0} \cap \mathcal{S}_{1,m_1}$ as intermediate quantities, and finally deriving $\mathcal{S}_{0,m_0} \cap \mathcal{S}_{1,m_1} \cap \mathcal{S}_{2,m_2}$. All these intersections are listed in Table 2. This proves that the proposed approach yields 8 intersections that are each restricted to one column of U and that this set of intersections yields all the individual columns of U, as required.

H.3: Pseudo-code

The above analysis also has the advantage of suggesting a possible structure for the software program used to implement this multi-stage QPT method. This program has a recursive structure, combined with a binary tree as follows. It is based on a recursive function, with each call of this function corresponding to one value of the stage index b_ℓ (and hence of b_r) and to one bit of the binary representation (H14) of the index of a column of U. In each such call, this function successively considers the two possible values of that bit, that is 0 and 1. For each of them, it determines the intersection of the subspaces with indices 0 to b_ℓ (by determining the subspace with index b_ℓ and combining it with the previously determined intersection of the subspaces with indices 0 to $(b_\ell - 1)$) and it recursively calls itself with this bit value. Successively performing this for each of the two values of the considered bit creates two branches in the tree structure. In each final call of this recursive function, i.e. each leave of the tree, all bits of the column index are assigned, the function determines the corresponding complete subspace intersection and sends it back as one column of the estimate of U. By gathering these columns at each

Table 2 Multi-stage QPT method

$\mathcal{S}_{0,m_0}$	$\mathcal{S}_{1,m_1}$	$\mathcal{S}_{2,m_2}$	$\mathcal{S}_{0,m_0} \cap \mathcal{S}_{1,m_1}$	$\mathcal{S}_{0,m_0} \cap \mathcal{S}_{1,m_1} \cap \mathcal{S}_{2,m_2}$
$\{1, 2, 3, 4\}$	$\{1, 2, 5, 6\}$	$\{1, 3, 5, 7\}$	$\{1, 2\}$	$\{1\}$
$\{1, 2, 3, 4\}$	$\{1, 2, 5, 6\}$	$\{2, 4, 6, 8\}$	$\{1, 2\}$	$\{2\}$
$\{1, 2, 3, 4\}$	$\{3, 4, 7, 8\}$	$\{1, 3, 5, 7\}$	$\{3, 4\}$	$\{3\}$
$\{1, 2, 3, 4\}$	$\{3, 4, 7, 8\}$	$\{2, 4, 6, 8\}$	$\{3, 4\}$	$\{4\}$
$\{5, 6, 7, 8\}$	$\{1, 2, 5, 6\}$	$\{1, 3, 5, 7\}$	$\{5, 6\}$	$\{5\}$
$\{5, 6, 7, 8\}$	$\{1, 2, 5, 6\}$	$\{2, 4, 6, 8\}$	$\{5, 6\}$	$\{6\}$
$\{5, 6, 7, 8\}$	$\{3, 4, 7, 8\}$	$\{1, 3, 5, 7\}$	$\{7, 8\}$	$\{7\}$
$\{5, 6, 7, 8\}$	$\{3, 4, 7, 8\}$	$\{2, 4, 6, 8\}$	$\{7, 8\}$	$\{8\}$

Each column of the table contains the indices of the column vectors that correspond to the considered subspace or subspace intersection defined in the first row of the table

level back from the recursion, the complete matrix that estimates U (up to phase factors at this stage) is obtained.

The corresponding pseudo-code is provided in Algorithm 3 for the main part of the program and Algorithm 4 for the recursive function.

Input : a) Set $\{\widehat{\rho}_{out,b_\ell}\}$ of estimates of output density matrices of process (provided by QST and obtained for input density matrices defined in Section 4): one estimate for each stage b_ℓ of recursion in Algorithm 4, b) Estimate $|\widehat{\Psi}_2\rangle$ of output ket $|\Psi_2\rangle$ (provided by QST and obtained for all components of input ket $|\Psi_1\rangle$ equal to $1/\sqrt{d}$).

Output: Estimate $\widehat{U}_5$ of quantum process matrix U.

begin

`/* Exploit` $\{\widehat{\rho}_{out,b_\ell}\}$`:` `*/`

$\widehat{U}_2$= EQPT5-sub(0, $\{\widehat{\rho}_{out,b_\ell}\}$,0);

`/* Exploit` $|\widehat{\Psi}_2\rangle$`:` `*/`

$|\Psi_3\rangle = \widehat{U}_2^\dagger |\widehat{\Psi}_2\rangle$;

$\widehat{U}_5 = \widehat{U}_2 \ \mathrm{diag}(|\Psi_3\rangle \oslash |\Psi_1\rangle)$;

end

Algorithm 3 EQPT5-main: main part of fifth Eigenanalysis-based QPT algorithm (composed of multiple stages). This version is for the case when the QST algorithm provides estimates that meet the known properties of the actual quantities. Otherwise, see text.

Input : a) Index b_ℓ of recursion stage, b) Set $\{\widehat{\rho}_{out,b_\ell}\}$ of estimates of output density matrices of process (see Algorithm 3), c) Matrix $\mathcal{I}$ whose column vectors form a basis of subspace intersection determined in previous stages of recursion (none in first stage).
Output: matrix $\widehat{U}_{part}$ related to estimation of part of quantum process matrix U.
begin
  /* Set auxiliary variables (using d: state space dimension; $d_1 = d/2$): */
  $b_{\ell m} = \log_2 d_1$;
  $b_n = 2^{b_\ell}$;
  $b_s = d_1/b_n$;
  /* Exploit single estimated output density matrix $\widehat{\rho}_{out,b_\ell}$ corresponding to current stage b_ℓ of recursion: */
  Eigendecomposition: derive 1) a diagonal matrix D_4 that contains the eigenvalues of $\widehat{\rho}_{out,b_\ell}$ in an arbitrary order and 2) a matrix V_4 whose columns are eigenvectors of $\widehat{\rho}_{out,b_\ell}$ in the same order as eigenvalues;
  Reorder the eigenvalues in D_4 in nonincreasing order and apply the same permutation to the columns of V_4 to create the matrix $\widehat{U}_{b_\ell}$;
  Store the first d_1 columns and last d_1 columns of $\widehat{U}_{b_\ell}$ respectively in matrices $\widehat{U}_{b_\ell}(1)$ and $\widehat{U}_{b_\ell}(2)$;
  /* Derive intersections $\mathcal{I}(1)$ and $\mathcal{I}(2)$ of subspaces defined by $\widehat{U}_{b_\ell}(1)$ and $\widehat{U}_{b_\ell}(2)$ with subspace defined by $\mathcal{I}$ (if any): */
  if $b_\ell = 0$ **then**
    $\mathcal{I}(1) = \widehat{U}_{b_\ell}(1)$;
    $\mathcal{I}(2) = \widehat{U}_{b_\ell}(2)$;
  end
  else
    $\mathcal{I}(1)$ = first b_s column vectors provided by CCA of $\mathcal{I}$ and $\widehat{U}_{b_\ell}(1)$;
    $\mathcal{I}(2)$ = first b_s column vectors provided by CCA of $\mathcal{I}$ and $\widehat{U}_{b_\ell}(2)$;
  end
  if $b_\ell < b_{\ell m}$ **then**
    $\widehat{U}_{part}(1)$ = EQPT5-sub $(b_\ell + 1, \{\widehat{\rho}_{out,b_\ell}\}, \mathcal{I}(1))$;
    $\widehat{U}_{part}(2)$ = EQPT5-sub $(b_\ell + 1, \{\widehat{\rho}_{out,b_\ell}\}, \mathcal{I}(2))$;
  end
  else
    /* If the CCA tool does not provide unit-norm vectors, then do it: */
    $\widehat{U}_{part}(1) = \mathcal{I}(1)/\text{norm}(\mathcal{I}(1))$;
    $\widehat{U}_{part}(2) = \mathcal{I}(2)/\text{norm}(\mathcal{I}(2))$;
  end
  $\widehat{U}_{part}$ = concatenation of columns of $\widehat{U}_{part}(1)$ and $\widehat{U}_{part}(2)$;
end

Algorithm 4 EQPT5-sub: recursively called sub-program of fifth Eigenanalysis-based QPT algorithm (composed of multiple stages). This version is for the case when the QST algorithm provides estimates that meet the known properties of the actual quantities. Otherwise, see text.

Appendix I Other performance criteria

I.1 Considered criteria

In Section 5, to analyze the error matrices $(U - e^{i\theta}\widehat{U}_5)$ with an optimized value of θ, we used a performance criterion based on the Frobenius norm of these error matrices. One may want to consider other performance criteria. In particular, with each vector norm, one may associate a corresponding matrix norm (see [50] p. 292) also called an operator norm (see [50] p. 294). This especially allows one to define the matrix 2-norm (see [78] p. 57) also called the spectral norm (see [50] p. 295). For a given matrix M, that norm may alternatively be defined as the square root of the largest eigenvalue of $M^\dagger M$ (see [50] p. 295, [78] p. 57) or as the largest singular value of M [82].

Whereas the concept of matrix 2-norm is thus completely defined, a problem then arises in its use that we have to make in the present paper, for the following reason. As in Sect. 5, we are here considering the error matrices $(U - e^{i\theta}\widehat{U}_5)$, and we are looking for the value of θ that yields the best fit for $\widehat{U}_5$ with respect to U up to a free phase factor. We are therefore looking for the value of θ that minimizes the considered norm of $(U - e^{i\theta}\widehat{U}_5)$, that is, here, its 2-norm (instead of its Frobenius norm in Sect. 5). Therefore,

as in Sect. 5, we do not initially consider a single matrix norm value but a continuum of matrix norm values, that are the norms of the set of matrices $(U - e^{i\theta}\widehat{U}_5)$ obtained when the angle θ is varied over the interval $[0, 2\pi[$, and we then aim at minimizing that type of norm with respect to θ. When we used the Frobenius norm in Sect. 5, we were able to derive the analytical expression of the optimum angle θ with respect to that Frobenius norm (see (36)). Therefore, in practice, we only had to compute a *single* value of the considered norm. In contrast, in the approach considered here, the problem we face is that we (that is, the authors of this paper) cannot provide an analytical expression of the angle θ that is optimum in the sense of *the matrix 2-norm* of $(U - e^{i\theta}\widehat{U}_5)$. Moreover, it seems quite unlikely that such an analytical expression might be derived by using the "standard approach" in the following sense. As explained above, this 2-norm is defined with respect to the largest eigenvalue of a matrix. Eigenvalues may be formally defined as the roots of the characteristic polynomial of the considered matrix. The degree of that polynomial is equal to the dimension of the considered matrix. The roots (and hence their largest value) of that polynomial cannot be analytically derived here, since the dimension of the considered square matrix is arbitrary.

To solve this problem, we therefore propose three approaches hereafter. All of them are based on the matrix 2-norm, so that their names start with "2norm". They differ in the value(s) of the angle θ used in the above-defined type of error matrices.

The first proposed performance criterion is defined as the 2-norm of the matrix $(U - e^{i\theta}\widehat{U}_5)$ with θ set to the value that is optimum in the sense of the Frobenius norm and that is defined in (36). We (authors) cannot guarantee whether the value in (36) is also optimum in the sense of the 2-norm (and our test results provided below clearly suggest it is not). Therefore, the value obtained with this first approach for the 2-norm of the error matrix is only considered as an approximation of the value that we are looking for. This approach, denoted as 2norm-Fro, has the advantage of being simple and of resulting in a low computation time. But it has the drawback of only providing an approximate value of the target 2-norm, with an undefined accuracy.

Our second approach consists of performing a sweep on the values of θ, computing the corresponding values of the 2-norm of the matrix $(U - e^{i\theta}\widehat{U}_5)$ and keeping the minimum of these values (as the value obtained for the best fit). More precisely, the considered values of θ are $2\pi k/N_s$, where $k = 0, \ldots, (N_s - 1)$, and N_s is the considered number of values of θ over the range $[0, 2\pi[$. N_s is a free parameter of the method and the accuracy of this method is expected to generally increase when N_s increases. In the tests reported below, this method, called 2norm-sweep, is considered to be of interest as compared with the 2norm-Fro method only when the value of the 2-norm that it yields is lower than or equal to the approximate value provided by the 2norm-Fro method (i.e. when 2norm-sweep yields a better fit than 2norm-Fro). The advantage of the 2norm-sweep method is that is can be made accurate by using a high enough value of N_s, but the price to pay to this end is its expected computational complexity, that may be justified as follows. First, even for a single value of N_s, the 2norm-sweep approach requires one to compute N_s values of matrix 2-norms (instead of only one for the 2norm-Fro method). Moreover, as explained above, we want to operate that 2norm-sweep method so that it meets a condition (namely, it should provide a lower 2-norm value than 2norm-Fro). To this end, we reapeatedly apply the 2norm-sweep method, with increasing values of N_s, until the above condition is met.

Our third method also computes the 2-norm of the matrix $(U - e^{i\theta}\widehat{U}_5)$ for several values of θ, but these values are then automatically selected by an algorithm, and this is expected to reduce the number of tested values of θ as compared with 2norm-sweep. More precisely, we here use the Nelder-Mead (NM) algorithm, as implemented in the "fminsearch" function of the Matlab software (with its default parameter values), with a cost function defined as the 2-norm of the matrix $(U - e^{i\theta}\widehat{U}_5)$, where U and $\widehat{U}_5$ are fixed, and θ is varied. The NM algorithm automatically varies θ so as to minimize the above cost function and provides its minimum value. The "fminsearch" Matlab function used here requires one to provide the initial value of θ, which should be set to a "guess" or approximation of the desired value. We here set it to the value that is optimum for the Frobenius norm and defined in (36). This third method is hereafter called 2norm-NM. As compared with 2norm-Fro, 2norm-NM has the advantage of being most probably more accurate, because it starts from the value of θ that is the solution of 2norm-Fro and then aims at refining it (for the 2-norm considered here). However, 2norm-NM is slower than 2norm-Fro, because it requires several evaluations of 2-norms. This also shows the usefulness of 2norm-Fro: even if one would only like to use 2norm-NM, as explained above, the 2norm-Fro method is also required, to provide the initial value of θ used in 2norm-NM. As compared with 2norm-sweep, 2norm-NM may be hoped to yield a lower computation time, by using a lower number of values of θ where 2-norms are computed, but this has to be experimentally confirmed, which is done further in this appendix. In contrast, 2norm-sweep may be used (when it is not too slow) to provide a "ground truth with controlled accuracy", since its accuracy is expected to increase with N_s, as explained above. This may be used, at least with some tests, to check that, as 2norm-NM operates with values of θ that are not controlled by the user, it succeeds in converging to a value of its 2-norm (i.e. cost function) that is similar to, or hopefully even lower than, the approximate but accurate value provided by 2norm-sweep for a "high" value of N_s (where the required range of values of N_s is derived from tests, as shown below).

I.2 Generic tests

I.2.1 Test conditions

Two types of tests were performed, in order to 1) first check and refine the above statements about the considered three methods for computing 2-norms, using a generic setup at this stage, and 2) then derive their performance when focusing on the application of these tools to QPT.

Table 3 Values of two performance parameters (vs. number of qubits), namely: 1) Execution times (per elementary test) of all three tested methods (2norm-Fro, 2norm-sweep, 2norm-NM) that compute performance criteria related to the 2-norm of the error matrix, and 2) Only for the 2norm-sweep method, minimum tested value of N_s (see text) for which 2norm-sweep yields better performance than 2norm-Fro

Method, parameter	Number of qubits			
	2	3	6	12
2norm-Fro, execution time	32 μs	36 μs	560 μs	9.4 s
2norm-sweep, execution time	1.1 s	0.46 s	0.20 s	1 h 14 mn
2norm-sweep, minimum N_s	200 000	50 000	500	500
2norm-NM, execution time	480 μs	580 μs	9.0 ms	2 mn 30 s

The first type of tests, corresponding to the above-mentioned generic setup, is reported here. In this first type of tests, each elementary test uses two matrices, that respectively correspond to U and $\widehat{U}_5$ in our final application to QPT, and that are therefore denoted in the same way hereafter, but that are in fact not yet actually derived from QPT in this first type of tests. U is a randomly drawn $d \times d$ matrix. Each of its elements is real-valued and randomly, uniformly, drawn in the range [0, 1]. $\widehat{U}_5$ is set to

$$\widehat{U}_5 = e^{i\pi/4}U + 0.01R \tag{I15}$$

when R is a random matrix drawn with the same dimension and statistics as U. In this first type of tests, this allows us to create a simple and realistic simulation of an estimate $\widehat{U}_5$ of U, without yet running a QPT algorithm, whereas such an algorithm will be actually used in our second type of tests, further in this appendix. In (I15), the phase factor $e^{i\pi/4}$ models the phase indeterminacy of QPT and R models estimation errors faced in QPT. For each considered value of d, 100 such elementary tests were performed, each with different, randomly drawn, matrices U and R. For each configuration, the execution time provided below is the mean, over all 100 elementary tests, of the execution time for a single elementary test. These execution times were measured for the same hardware and software environment as in Sect. 5. Moreover, as in Sect. 5, the above-defined protocol was repeated for several values of the state space dimension d that take the form $d = 2^q$, where q is the equivalent number of qubits. Here, only a few values of q are considered, especially in order to determine the general trend of the dependence of the execution time with respect to q.

I.2.2 Test results

Based on the above discussion, the 2norm-sweep method is here stated to "succeed" in a given configuration if it yields a 2-norm (of the error matrix) lower than or equal to that of the 2norm-Fro method for each of the 100 elementary tests performed in that configuration. Otherwise, it it stated to "fail" in that configuration. Table 3 shows results obtained in the above-defined conditions. For the 2norm-sweep method, the execution time reported in Table 3 is the one corresponding to the minimum value of N_s needed for 2norm-sweep to succeed, when only testing values of N_s equal to 1, 2 and 5 in each decade, that is, e.g., 100, 200, 500, 1000, 2000, 5000 and so on.

For this first type of tests, Table 3 and our other test results yield the following conclusions about all three tested methods, from the point of view of their execution time and accuracy. Table 3 first shows that, in order to succeed, 2norm-sweep requires an execution time that is always much higher than those of 2norm-Fro and 2norm-NM, and even unacceptably high in some configurations. In particular, for $q = 12$ qubits, the total execution time for applying 2norm-sweep to all 100 elementary tests is higher than 5 days. Moreover, this execution time is obtained when only considering the value of N_s that is finally kept (and provided in Table 3). But, to determine that value, one must moreover repeat this processing for various values of N_s, which increases the overall execution time. Besides, without prior knowledge, N_s must be varied in a wide range, since the adequate value of N_s highly depends on the considered value of q, as shown by Table 3.

Moreover, when paying the price of using 2norm-sweep with values of N_s that are high enough to make that 2norm-sweep method succeed, it does not yield any significant

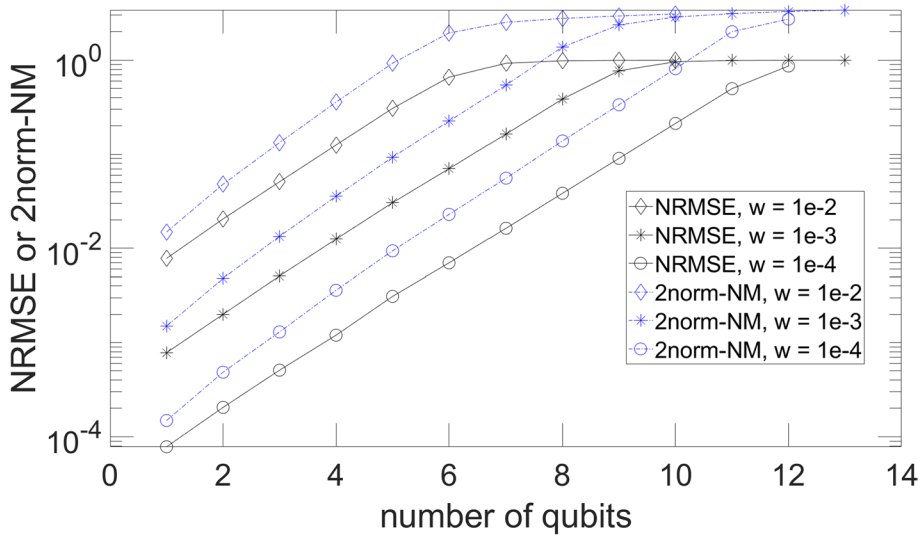

Fig. 3 Normalized Root Mean Square Error (NRMSE, black solid lines) and 2-norm of error matrix (blue dash doted lines), when estimating the process matrix U with the proposed Quantum Process Tomography (QPT) method called EQPT1, vs. number q of qubits, for several values of the parameter w that defines the magnitude of the estimation errors of Quantum State Tomography (QST)

improvement in terms of accuracy as compared with the 2norm-NM method. More precisely, for some of the 100 elementary tests, 2norm-sweep yields a lower 2-norm than the 2norm-NM method, but anyway the relative difference[6] between their 2-norms is very low, since it spans the range $[-4 \times 10^{-5}, 2 \times 10^{-5}]$ for $q = 2$ or 3 and $[-8 \times 10^{-3}, -2 \times 10^{-3}]$ for $q = 6$ or 12, over all 100 elementary tests and considered values of N_s. Still in the case when it succeeds, 2norm-sweep yields a lower 2-norm than 2norm-Fro for all 100 elementary tests by definition, but the relative difference[7] between their 2-norms is limited, since it spans the range $[-4.5\%, 0]$, over all 100 elementary tests and considered configurations (i.e. values of N_s and q).

It may also be noted that, when it fails, 2norm-sweep yields very poor performance. On each elementary test, it turns out to yield a higher 2-norm than the 2norm-Fro and 2norm-NM methods. Moreover, the relative difference between the 2-norms provided by 2norm-sweep and 2norm-Fro is very high, since it spans the range $[12\%, 28\%]$, over all 100 elementary tests and considered configurations (i.e. values of N_s and q). The relative difference between the 2-norms provided by 2norm-sweep and 2norm-NM is even higher, since it spans the range $[-41\%, -15\%]$, over all 100 elementary tests and considered configurations (i.e. values of N_s and q).

Based on the above-mentioned results about execution time and accuracy, the 2norm-NM method appears to be much more attractive than 2norm-sweep. Moreover, the results obtained with 2norm-NM seem to be safe, based on their comparison to those provided by 2norm-sweep: when operating 2norm-sweep with a high value of N_s, it is expected to yield an accurate value of the actual 2-norm to be estimated; since 2norm-NM turns out to yield almost the same performance as 2norm-sweep in that case, this suggests than the NM algorithm does succeed in converging to a relevant solution in all our tests.

[6] This relative difference is defined as "(2norm-NM - 2norm-sweep) / 2norm-NM", where "2norm-sweep"' and "2norm-NM" are the values of the 2-norm provided by our considered two methods.

[7] This relative difference is defined as "(2norm-sweep - 2norm-Fro) / 2norm-sweep", where "2norm-Fro"' and "2norm-sweep" are the values of the 2-norm provided by our considered two methods.

Now, as compared with 2norm-Fro, the 2norm-NM method always yields a lower 2-norm and the relative difference[8] between these norms spans the range $[-4.5\%, 0]$. The 2norm-NM method is therefore somewhat more accurate than 2norm-Fro, and safer because it aims at estimating the actual 2-norm of interest, whereas 2norm-Fro is not guaranteed to estimate it. More precisely, the difference between the 2-norms respectively obtained with 2norm-Fro and 2norm-NM is not negligible, which tends to confirm that the value of θ that is optimum with respect to the Frobenius norm of the error matrix (see (36)) and used in the 2norm-Fro method is actually *different* from the value of θ that we aim at estimating here, that is estimated by the 2norm-NM method and that is optimum with respect to the 2-norm of the error matrix. The execution time required by 2norm-NM to achieve the above features is significantly higher than that of 2norm-Fro but remains acceptable, as shown by Table 3.

As the final conclusion for this first type of tests, our preferred method (for the considered numbers of qubits) is therefore 2norm-NM (and one should keep in mind that it uses 2norm-Fro for its initialization). Instead, the 2norm-Fro method could be used alone, to decrease computation time but at the expense of providing a less accurate solution. Finally, the 2norm-sweep method is not recommended. Therefore, in the application of these tools to QPT reported hereafter, we put the emphasis on 2norm-NM and we provide some additional information about 2norm-Fro, but we do not consider 2norm-sweep.

I.3 Application to QPT

We here use the same test conditions as in Sect. 5, and we consider the EQPT1 method, as an example. The performance thus obtained in terms on NRMSE is therefore the same as in Sect. 5. In addition, we mainly compare it to the values of 2-norms provided by our 2norm-NM method. The corresponding results are provided in Fig. 3. The main conclusion to be drawn from this figure is that the curves respectively associated with NRMSE and 2-norm (as computed by 2norm-NM) for any given value of w are more or less parallel in this semilogarithmic representation (and rather close), which means that the ratio between these two parameters does not vary much over the wide range of variation of these parameters (and that this ratio is not very high). Therefore, the 2-norm parameter does not bring much information in addition to the information already provided by NRMSE.

In addition, our 2norm-Fro and 2norm-NM methods here compare as follows. Whatever the values of w and of the number q of qubits, for each of the 100 elementary tests, the 2-norm provided by 2norm-NM is lower than or equal to the 2-norm provided by 2norm-Fro. This again tends to show that the Nelder-Mead (NM) algorithm converged to a relevant value and, at least, this proves that it always performs better than 2norm-Fro in the considered tests. Here again, 2norm-NM is therefore the preferred method. Moreover, the relative difference between the results of 2norm-Fro and 2norm-NM is in the range $[-24\%, 0]$. There exist various cases where 2norm-NM yields a quite significantly lower 2-norm than 2norm-Fro, which is an additional motivation for preferably using 2norm-NM.

[8] This relative difference is defined as "(2norm-NM - 2norm-Fro) / 2norm-NM", where "2norm-Fro"' and "2norm-NM" are the values of the 2-norm provided by our considered two methods.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s44464-026-00030-y.

Supplementary Material 1.

Author contributions

Y.D. wrote the main manuscript text and prepared the figures. Y.D. and A.D. reviewed the manuscript.

Funding

No funding was received for conducting this study.

Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing of interests

The authors have no relevant financial or non-financial interests to disclose. The authors have no conflict of interest to declare that are relevant to the content of this article.

Received: 30 January 2026 / Revised: 23 June 2026 / Accepted: 7 July 2026

Published online: 28 July 2026

References

1. Nielsen MA, Chuang IL. Quantum computation and quantum information. Cambridge, UK: Cambridge University Press; 2000.
2. Chuang IL, Nielsen MA. Prescription for experimental determination of the dynamics of a quantum black box. J Mod Opt. 1997;44(11–12):2455–67.
3. Baldwin CH, Kalev A, Deutsch I. Quantum process tomography of unitary and near-unitary maps. Phys Rev A. 2014;90:012110–101211010.
4. Blume-Kohout R, Gamble JK, Nielsen E, Mizrahi J, Sterk JD, Maunz P. Robust, self-consistent, closed-form tomography of quantum logic gates on a trapped ion qubit. arXiv:1310.4492v1. 2013.
5. Branderhorst MPA, Nunn J, Walmsley IA, Kosut RL. Simplified quantum process tomography. New J Phys. 2009;11:115010–2.
6. Merkel ST, Gambetta JM, Smolin JA, Poletto S, Córcoles AD, Johnson BR, et al. Self-consistent quantum process tomography. Phys Rev A. 2013;87:062119–10621199.
7. Navon N, Akerman N, Kotler S, Glickman Y, Ozeri R. Quantum process tomography of a Mølmer-Sørensen interaction. Phys Rev A. 2014;90:010103–10101035.
8. Shukla A, Mahesh TS. Single-scan quantum process tomography. Phys Rev A. 2014;90:052301–10523016.
9. Takahashi M, Bartlett SD, Doherty AC. Tomography of a spin qubit in a double quantum dot. Phys Rev A. 2013;88:022120–10221209.
10. Wang Y, Dong D, Petersen IR, Zhang J. An approximate algorithm for quantum Hamiltonian identification with complexity analysis. In: Proceedings of the 20th world congress of the international federation of automatic control (IFAC 2017). Toulouse, France; 2017. p. 12241–5.
11. White AG, Gilchrist A. Measuring two-qubit gates. J Opt Soc Am B. 2007;24(2):172–83.
12. Shabani A, Kosut RL, Mohseni M, Rabitz H, Broome MA, Almeida MP, et al. Efficient measurement of quantum dynamics via compressive sensing. Phys Rev Lett. 2011;106(10(Mar.)):100401–4.
13. Kimmel S, Liu Y-K. Phase retrieval using unitary 2-designs. In: 2017 International conference on sampling theory and applications (SampTA). Tallinn, Estonia; 2017. p. 345–9.
14. Fiurasek J, Hradil Z. Maximum-likelihood estimation of quantum processes. Phys Rev A. 2001;63(2):020101–4.
15. Sakhouf H, Daoud M, Ahl-Laamara R. Quantum process tomography of the single-shot entangling gate with superconducting qubits. J Phys B: At Mol Opt Phys. 2023;56(10):105501.
16. Xiao S, Wang Y, Dong D, Zhang J. Two-stage solution for ancilla-assisted quantum process tomography: error analysis and optimal design. In: 2023 62nd IEEE conference on decision and control (CDC). 2023. p. 7178–83.
17. Stanchev SG, Vitanov NV. Multipass quantum process tomography. Sci Rep. 2024;14:18185.
18. Xiao S, Wang Y, Zhang J, Dong D, Mooney G-J, Petersen I-R, et al. A two-stage solution to quantum process tomography: error analysis and optimal design. IEEE Trans Inf Theory. 2025;71(3):1803–23.
19. Antesberger M, Quintino MT, Walther P, Rozema LA. Higher-order process matrix tomography of a passively-stable quantum switch. PRX Quantum. 2024;5(1):010325–22.
20. Dobrynin D, Cardarelli L, Muller M, Bermudez A. Compressed-sensing Lindbladian quantum tomography with trapped ions. 2024. https://arxiv.org/abs/2403.07462
21. Xue S, Huang G, Liu Y, Wang D, Shi W, Liu Y, et al. Efficient quantum process tomography for Clifford circuits. Phys Rev A. 2023;108(3):032419–9.

22. Verdeil F, Deville Y. Unitary quantum process tomography with unreliable pure input states. Phys Rev A. 2023;108(6):062410–20.
23. Xue S, Liu Y, Wang Y, Zhu P, Guo C, Wu J. Variational quantum process tomography of unitaries. Phys Rev A. 2022;105(3):032427–8.
24. Gutoski G, Johnston N. Process tomography for unitary quantum channels. J Math Phys. 2014;55(3):032201–19.
25. Gutierrez IL, Dietrich F, Mendl C. Quantum process tomography of unitary maps from time-delayed measurements. In: Quantum information processing, vol. 22. 2023. p. 251.
26. Escandón-Monardes J, Uzcátegui D, Rivera-Tapia M, Walborn S, Delgado A. Estimation of high-dimensional unitary transformations saturating the Quantum Cramér-Rao bound. Quantum. 2024;8:1405.
27. Zhao H, Lewis L, Kannan I, Quek Y, Huang H-Y, Caro M. Learning quantum states and unitaries of bounded gate complexity. PRX Quantum. 2024;5(4):040306.
28. IBM: bringing useful quantum computing to the world. https://www.ibm.com/quantum (accessed on line on Nov. 22, 2024)
29. IBM: Qiskit. https://www.ibm.com/quantum/qiskit (accessed on line on Nov. 22, 2024)
30. IBM: circuit library. https://docs.quantum.ibm.com/ guides/circuit-library (accessed on line on Nov. 22, 2024)
31. IBM: Circuit library / standard gates. https://docs.quantum.ibm.com/api/qiskit/ circuit_library#standard-gates (accessed on line on Nov. 22, 2024)
32. Amazon: amazon braket terms and concepts. https://docs.aws.amazon.com/braket/latest/ developerguide/braket-terms.html (accessed on line on Nov. 22, 2024)
33. Mathworks: introduction to quantum computing. https://fr.mathworks.com/help/matlab/math/ introduction-to-quantum-computing.html (accessed on line on Nov. 22, 2024)
34. Odake T, Yoshida S, Murao M. Analytical lower bound on the number of queries to a black-box unitary operation in deterministic exact transformations of unknown unitary operations. 2024. https://arxiv.org/abs/2405.07625
35. ...Abbas A, Ambainis A, Augustino B, Bärtschi A, Buhrman H, Coffrin C, et al. Challenges and opportunities in quantum optimization. Nat Rev Phys. 2024;6(December):718–35.
36. Deville Y, Duarte LT, Deville A. A taxonomy of relationships between information processing, machine learning and quantum physics: quantum-inspired, quantum-assisted, quantum-targeted and related approaches. In: Proceedings of the 2024 IEEE mediterranean and middle-east geoscience and remote sensing symposium (M2GARSS 2024), Oran, Algeria, 2024;361–365
37. Deville Y, Deville A. New single-preparation methods for unsupervised quantum machine learning problems. IEEE transactions on quantum engineering 2(article sequence number: 3104224, https://doi.org/10.1109/TQE.2021.3121797),1–24 2021
38. Deville Y, Deville A. From blind quantum source separation to blind quantum process tomography. In: Proceedings of the 12th international conference on latent variable analysis and signal separation (LVA/ICA 2015), Liberec, Czech Republic, Springer International Publishing Switzerland, LNCS 9237, 2015;184–192
39. Deville Y, Deville A. The blind version of quantum process tomography: operating with unknown input values. In: Proceedings of the 20th world congress of the international federation of automatic control (IFAC 2017), Toulouse, France, 2017;12228–12234
40. Deville Y, Deville A. Quantum process tomography with unknown single-preparation input states: concepts and application to the qubit pair with internal exchange coupling. Phys Rev A. 2020;101(4):042332–104233218.
41. Deville Y, Deville A. Blind separation of quantum states: estimating two qubits from an isotropic Heisenberg spin coupling model. In: Proceedings of the 7th international conference on independent component analysis and signal separation (ICA 2007), ISSN 0302-9743, Springer-Verlag, Vol. LNCS 4666. Erratum: replace two terms $E\{r_i\}E\{q_i\}$ in (33) of [41] by $E\{r_i q_i\}$, since q_i depends on r_i., London, UK, 2007;706–713
42. Deville Y, Deville A. Exploiting the higher-order statistics of random-coefficient pure states for quantum information processing. Quantum Inf Process. 2023;22:216.
43. Mathworks: Sparse matrices. https://www.mathworks.com/help/matlab/sparse-matrices.html (accessed on line on June 25, 2024)
44. Shabani A, Mohseni M, Lloyd S, Kosut RL, Rabitz H. Estimation of many-body quantum Hamiltonians via compressive sensing. Phys Rev A. 2011;84(1(Jul.)):012107–8.
45. Reich DM, Gualdi G, Koch CP. Minimum number of input states required for quantum gate characterization. Phys Rev A. 2013;88(4):042309–9.
46. Messiah A. Mécanique Quantique - 1. Paris: Dunod; 1962.
47. Feynman RP. Statistical Mechanics. Reading (Massachusets): a set of lectures. W. A. Benjamin Inc.; 1972.
48. Fano U. Description of states in quantum mechanics by density matrix and operator techniques. Rev Mod Phys. 1957;29(1):74–93.
49. Neumann J. Mathematical foundations of quantum mechanics (translated from the German edition). Princeton: Princeton University Press; 1955.
50. Horn RA, Johnson CR. Matrix analysis. Cambridge, United Kingdom: Cambridge University Press; 1985.
51. Horn RA, Johnson CR. Matrix analysis. Cambridge, United Kingdom: Cambridge University Press; 2013.
52. Hyvarinen A, Karhunen J, Oja E. Independent component analysis. New York: Wiley; 2001.
53. Papoulis A. Probability, random variables, and stochastic processes. Singapore: McGraw-Hill; 1984.
54. Scharf LL. Statistical signal processing. Detection: estimation and time series analysis. Addison-Wesley, Reading, Massachusetts; 1991.
55. Smilde A, Bro R, Geladi P. Multi-way analysis with applications in the chemical sciences. Chichester, England: Wiley; 2004.
56. Saporta G. Probabilités. Paris: Analyse des Données et Statistique. Technip; 1990.
57. Mathworks: Canonical correlation analysis. https://fr.mathworks.com/help/stats/canoncorr.html (accessed on line on March 4, 2024)
58. Boulais A, Deville Y, Berné O. A blind identification and source separation method based on subspace intersections for hyperspectral astrophysical data. In: 13th international conference on latent variable analysis and signal separation (LVA/ICA 2017), vol. 10169. LNCS. Grenoble, France: Springer International Publishing AG; 2017;367–80.

59. Deville Y, Duarte LT, Hosseini S. Nonlinear blind source separation and blind mixture identification. Methods for bilinear, linear-quadratic and polynomial mixtures. Springer, 2021
60. Deville Y. Blind source separation and blind mixture identification methods. Wiley encyclopedia of electrical and electronics engineering, pp. 1–33. Wiley, J. Webster (ed.), 2016
61. DeLathauwer L, Castaing J, Cardoso J-F. Fourth-order cumulant-based blind identification of underdetermined mixtures. IEEE Trans Signal Process. 2007;55(6):2965–73.
62. Tong L, Liu R, Soon VC, Huang Y-F. Indeterminacy and identifiability of blind identification. IEEE Trans Circuits Syst. 1991;38(5):499–509.
63. Lvovsky AI, Raymer MG. Continuous-variable optical quantum-state tomography. Rev Mod Phys. 2009;81(1):299–332.
64. Medford J, Beil J, Taylor JM, Bartlett SD, Doherty AC, Rashba EI, et al. Self-consistent measurement and state tomography of an exchange-only spin qubit. Nat Nanotechnol. 2013;8(Sept.):654–9.
65. Stark C. Self-consistent tomography of the state-measurement gram matrix. Phys Rev A. 2014;89:052109–10521098.
66. James DFV, Kwiat PG, Munro WJ, White AG. Measurement of qubits. Phys Rev A. 2001;64:052312–5.
67. Roth I, Wilkens J, Hangleiter D, Eisert J. Semi-device-dependent blind quantum tomography. Quantum. 2023;7(Jul.):1053.
68. Melkani A, Gneiting C, Nori F. Eigenstate extraction with neural-network tomography. Phys Rev A. 2020;102(2):022412–102241211.
69. Ahmed S, Munoz CS, Nori F, Kockum AF. Quantum state tomography with conditional generative adversarial networks. Phys Rev Lett. 2021;127(14(Sep.)):140502–8.
70. Ahmed S, Munoz CS, Nori F, Kockum AF. Classification and reconstruction of optical quantum states with deep neural networks. Phys Rev Res. 2021;3(3(Sep.)):033278.
71. Altepeter JB, Jeffrey ER, Kwiat PG. Photonic state tomography. Adv At Mol Opt Phy. 2005;52:105–59.
72. Flammia ST, Gross D, Liu Y-K, Eisert J. Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators. New J Phys. 2012;14:095022–8.
73. Dong D, Wang Y, Hou Z, Qi B, Pan Y, Xiang G-Y. State tomography of qubit systems using linear regression estimation and adaptive measurements. In: Proceedings of the 20th world congress of the international federation of automatic control (IFAC 2017). Toulouse, France; 2017. p. 13556–61.
74. Kurmapu MK, Tiunova VV, Tiunov ES, Ringbauer M, Maier C, Blatt R, et al. Reconstructing complex states of a 20-qubit quantum simulator. PRX Quantum. 2023;4(4):040345–9.
75. Stricker R, Meth M, Postler L, Edmunds C, Ferrie C, Blatt R, et al. Experimental single-setting quantum state tomography. PRX Quantum. 2022;3:040310–34.
76. Verdeil F, Deville Y. Pure-state tomography with parallel unentangled measurements. Phys Rev A. 2023;107(1):012408–15.
77. Cotler J, Wilczek F. Quantum overlapping tomography. Phys Rev Lett. 2020;124(10):100401–6.
78. Golub GH, Loan CFV. Matrix Computations. Baltimore: Johns Hopkins University Press; 1996.
79. Mathworks: QR decomposition. https://fr.mathworks.com/help/matlab/ ref/qr.html (accessed on line on May 22, 2025)
80. Horn RA, Johnson CR. Topics in matrix analysis. Cambridge, United Kingdom: Cambridge University Press; 1999.
81. Peres A. Quantum theory: concepts and methods. Dordrecht, The Netherlands: Kluwer Academic; 1995.
82. Mathworks: vector and matrix norms. https://fr.mathworks.com/help/matlab/ref/ norm.html?lang=en (accessed on line on Dec. 16, 2025)

Publisher's Note