

Quantum Science and Technology



OPEN ACCESS

PAPER

Graph test of controllability in qubit arrays: a systematic way to determine the minimum number of external controls

RECEIVED
17 January 2023REVISED
6 June 2023ACCEPTED FOR PUBLICATION
26 June 2023PUBLISHED
10 July 2023

Fernando Gago-Encinas , Monika Leibscher and Christiane P Koch*

Fachbereich Physik and Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, Arnimallee 14, 14195 Berlin, Germany

* Author to whom any correspondence should be addressed.

E-mail: christiane.koch@fu-berlin.de**Keywords:** quantum control, universal quantum computing, controllability analysis, superconducting qubits

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Abstract

The ability to implement any desired quantum logic gate on a quantum processing unit is equivalent to evolution-operator controllability of the qubits. Conversely, controllability analysis can be used to minimize the resources, i.e. the number of external controls and qubit–qubit couplings, required for universal quantum computing. Standard controllability analysis, consisting in the construction of the dynamical Lie algebra, is, however, impractical already for a comparatively small number of qubits. Here, we show how to leverage an alternative approach, based on a graph representation of the Hamiltonian, to determine controllability of arrays of coupled qubits. We provide a complete computational framework and exemplify it for arrays of five qubits, inspired by the *ibmq_quito* architecture. We find that the number of controls can be reduced from five to one for complex qubit–qubit couplings and to two for standard qubit–qubit couplings.

1. Introduction

Universal quantum computing requires evolution-operator controllability on the quantum processing units in order to perform every possible quantum logic gate. One way to achieve this, pursued for example in most superconducting qubit architectures, is to couple each qubit to at least one other qubit and drive all qubits locally. This approach becomes impractical for larger qubit arrays due to increasing requirements on physical space for control lines and on calibration time for control pulses. One may wonder whether a number of local drives smaller than the number of qubits would already be sufficient. If so, this would suggest the possibility of more resource-efficient architectures than currently in use.

Here, we show that controllability analysis provides a systematic approach to determine the minimum number of local controls for which any desired quantum logic gate can be implemented. We find that indeed the number of local controls can be smaller than the number of qubits. The minimum number of local controls, for a given size of the qubit array, depends on the type of qubit–qubit couplings. To facilitate analysis of medium-sized qubit arrays, we leverage a graph theory-based approach to controllability analysis and investigate five-qubit arrays starting from the *ibmq_quito* architecture.

Controllability analysis answers the question, in a yes–no fashion, which states can be reached by time evolution from a set of initial states [1, 2]. The standard approach consists in determining the rank of the dynamical Lie algebra of the system and comparing it to the dimension of the algebra that generates the unitary group of all time evolutions [3]. This approach has been extensively used to prove controllability for finite-dimensional systems with sufficiently small Hilbert space dimension [4–6] and can also be employed in subspaces of infinite-dimensional systems, when combined with Galerkin-type approximations [7–10]. The focus of these studies has been on single quantum systems. For multi-partite systems, where the Hilbert space dimension scales exponentially in the number of subsystems, evaluating the Lie rank condition quickly becomes challenging. It has thus mainly been used to identify controllable subspaces in systems which as a whole are not controllable [11–14]. Positive controllability results for qubit arrays, as needed for universal quantum computing, require an alternative approach.

The exponential scaling of Hilbert space dimension with the number of qubits is evidently inherent to gate-based quantum computing and represents a fundamental obstacle to controllability analysis that cannot be overcome with classical computers. But even for qubit arrays which are classically simulable, use of the Lie rank condition is hampered by numerical instabilities which are common when constructing orthogonal bases in large operator spaces. This latter obstacle can be avoided by resorting to graph theory-based methods for controllability analysis [9, 15, 16] which have successfully been applied to quantum walks [17], quantum networks [18], and quantum rotors [10, 19]. The latter are characterized by a highly degenerate spectrum which results in multiple resonant transitions, i.e. transitions with the same resonance frequency that are driven by the same external control. Controllability can then not simply be read off from the connectivity of the graph. This problem is also present for arrays of locally driven coupled qubits but the different spectral structure requires different graphical methods to prove controllability, as we will discuss below.

The manuscript is organized as follows. The basic concepts of controllability analysis are briefly reviewed in section 2. The methodology we suggest to use for controllability analysis of coupled qubit arrays is presented in section 3 with section 3.1 explaining how to deal with resonant transitions, section 3.2 presenting the actual graph test in the form of a flow chart and three algorithms and section 3.3 illustrating the use of the algorithms on simple two-qubit examples. Our results on five-qubit arrays inspired by the *ibmq_quito* architecture are presented in section 4, and section 5 concludes.

2. Controllability analysis

We study quantum systems that couple linearly to controls, such that their Hamiltonian can be expressed as

$$\hat{H}(t) = \hat{H}(t; u_1, \dots, u_m) = \hat{H}_0 + \sum_{j=1}^m u_j(t) \hat{H}_j, \quad (1)$$

where the controls $u_j(t)$ are real-valued functions and \hat{H}_j are the control operators. A state that evolves under equation (1) is given by $|\psi(t)\rangle = \hat{U}(t; u_1, \dots, u_m)|\psi(0)\rangle$. We consider coherent evolutions because it is the relevant foundation for gate-based quantum information.

For qubit arrays, the drift \hat{H}_0 is split into $\hat{H}_0 = \sum_{l=1}^{N_Q} \hat{H}_{0,l} + \hat{H}_c$, representing the independent local Hamiltonians $\hat{H}_{0,l}$ of the N_Q free qubits and the time-independent couplings \hat{H}_c between them. Typically, local controls \hat{H}_j act on single qubits, for example in the form of laser pulses in ion arrays [20] or varying microwave fields for certain superconducting qubits [21]. Regardless of the physical implementation of the qubits, we would like to answer the question whether a system with a set of controls is capable of performing any unitary operation.

The answer can be found by analyzing the controllability; in particular, the evolution-operator controllability of the system [3]. It is defined as follows:

Definition 1. A system with controls as defined in equation (1) and Hilbert space dimension n is **evolution-operator controllable** if and only if for every unitary evolution $\hat{U}_{\text{target}} \in U(n)$ there exist a final time $T \geq 0$, a phase angle $\theta \in [0, 2\pi]$ and a set of controls $\{u_j\}_{j=1}^m$ such that $|\psi(T)\rangle = e^{i\theta} \hat{U}_{\text{target}} |\psi(0)\rangle = \hat{U}(t; u_1, \dots, u_m) |\psi(0)\rangle$ for any initial state $|\psi(0)\rangle$ in the system's Hilbert space.

Note that $e^{i\theta}$ is a global phase that does not depend on the choice of the initial state. In other words, a system is evolution-operator controllable if and only if we can always choose controls and a final time to carry out any unitary operation, up to a global phase. For simplicity, we refer to this property as ‘controllable’.

A widely used method for studying the controllability of a quantum system is to analyze the **dynamical Lie algebra** [3, 17, 22]. It is generated by the drift and the m control operators of the system, $\mathcal{L} := \text{Lie}(i\hat{H}_0, i\hat{H}_1, \dots, i\hat{H}_m)$, i.e. it contains the skew-Hermitian operators $i\hat{H}_0, i\hat{H}_1, \dots, i\hat{H}_m$, and their (nested) commutators. A system with Hilbert space dimension n is controllable if the dimension of the dynamical Lie algebra is n^2 or $n^2 - 1$. In other words, it is controllable if the dimension of the Lie algebra matches either that of $\mathfrak{su}(n)$ (which generates the special unitary group $SU(n)$) or $\mathfrak{u}(n)$ (which generates the unitary group $U(n)$). This is sometimes referred to as the Lie algebra rank condition [3]. The main difference between generating $U(n)$ or $SU(n)$ is that the former allows the system to perform any unitary evolution including all global phases. Conversely, the latter encompasses all unitary evolutions up to a non-controllable global phase.

Alternatively to constructing the dynamical Lie algebra, a graph test can be used to analyze controllability [10, 16, 23]. The graph encodes the information of the Hamiltonian.

Definition 2. Given a quantum system evolving under equation (1), we can construct an undirected graph according to the rules:

- (i) For every eigenstate $|e_k\rangle$ of \hat{H}_0 , add a vertex to the graph with a corresponding label.
- (ii) For every control \hat{H}_j and every nonzero element $\langle e_i | \hat{H}_j | e_k \rangle$, add an edge between the vertices labelled $|e_i\rangle$ and $|e_k\rangle$. Additionally, add a label to every edge stating which control drives that transition.

We call this the **graph of a quantum system**¹.

Since the nodes of the graph are labelled by the eigenstates $|e_k\rangle$, we use these labels also to name the edges that represent the transitions generated by the controls. The edge joining the vertices $|e_a\rangle$ and $|e_b\rangle$ is denoted by (a, b) . For Hermitian controls, the transitions will necessarily happen in both directions, implying an undirected graph. This means, in particular, that both (a, b) and (b, a) refer to the same edge.

An important aspect of the graph are the energy gaps related to the graph edges $\Delta E_{a,b} := |E_{|e_a\rangle} - E_{|e_b\rangle}|$. If two transitions have the same energy gap $\Delta E_{a,b}$, they are called ‘resonant’. Resonant transitions may pose a challenge to controllability since, depending on the local controls, it may be impossible to address them independently.

Definition 3. Let \mathcal{G} be the graph of a system. If two edges $(a, b), (c, d) \in E(\mathcal{G})$ belong to the same control \hat{H}_j and have degenerate energy gaps $\Delta E_{a,b} = \Delta E_{c,d}$, then the two transitions represented by the edges are **coupled** to one another. Alternatively, if an edge $(a, b) \in E(\mathcal{G})$ is not coupled to any other transition belonging to its own control \hat{H}_j , the transition is said to be **decoupled**.

This leads to the following theorem whose proof can be found in [7]:

Theorem 1. *Let \mathcal{S} be a quantum system following equation (1) and let \mathcal{G} be its associated graph. The system \mathcal{S} is controllable if there exists a connected subgraph of \mathcal{G} that contains all vertices of \mathcal{G} and only decoupled transitions.*

An undirected graph is said to be connected if for every two vertices there exists a chain of adjacent edges that creates a path between the two selected vertices. A subgraph of a graph \mathcal{G} is a graph defined by a subset of the vertices $V(\mathcal{G})$ and a subset of the edges $E(\mathcal{G})$ that only link vertices in the subset. This means that a subgraph is a graph that we obtain by removing any number of the edges and vertices from the original graph.

The main benefit of the graph test in theorem 1 is that it allows us to avoid calculating the full dynamical Lie algebra. The graph of a quantum system with Hilbert space dimension n has n vertices. To have a connected subgraph we need to find $n - 1$ decoupled transitions such that they create a single connected component. This works quite efficiently if there are no resonant transitions in the system.

In the following, we expand this graph test such that it also works with resonant transitions. To this end, we adapt graphical methods for controllability analysis in degenerate systems [19] to systems with degenerate energy gaps driven by multiple controls. Since resonant transitions are very common in qubit arrays due to the multi-partite structure of the system, this modification opens the route to efficient analysis of controllability of qubit arrays.

3. Graph test of controllability for coupled subsystems

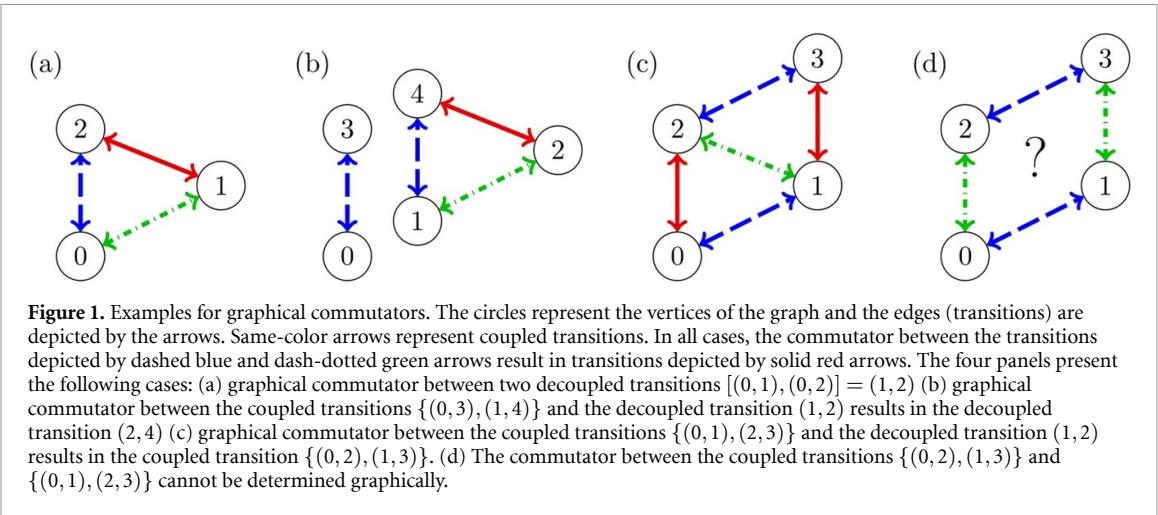
In this section we outline the algorithm for a graph test for coupled subsystems with resonant transitions. In section 3.1 we present methods to determine the graphical commutators already used in rotor systems [10] and we introduce the concept of subalgebras to treat resonant transitions driven by multiple controls. We show in section 3.2 that the graphical commutators can be calculated in a systematic way, avoiding the construction of the entire dynamical Lie algebra of a system. We illustrate the use of these methods in section 3.3 with simple two-qubit examples.

3.1. Resonant transitions and graphical commutators

We first present two concepts to decouple resonant transitions without calculating the complete Lie algebra of the system. To this end, we introduce the generalized skew-Hermitian Pauli matrices

$$\begin{aligned}\hat{G}_{j,k} &= e_{j,k} - e_{k,j}, \\ \hat{F}_{j,k} &= ie_{j,k} + ie_{k,j}, \\ \hat{D}_{j,k} &= ie_{j,j} - ie_{k,k},\end{aligned}\tag{2}$$

¹ Note that given two vertices there can be more than one edge between the two of them (belonging to different controls). Technically, this makes the graph of a system a multigraph.



where $e_{j,k}$ is the null matrix except for a 1 in the entry (j,k) . The commutators of these matrices with the skew-Hermitian drift are given by [19, 23]

$$\begin{aligned} [i\hat{H}_0, \hat{G}_{j,k}] &= (E_j - E_k) \hat{F}_{j,k}, \\ [i\hat{H}_0, \hat{F}_{j,k}] &= - (E_j - E_k) \hat{G}_{j,k}. \end{aligned} \quad (3)$$

Consider a transition (i, k) with transition matrix element

$$\langle e_i | \hat{H}_j | e_k \rangle = \alpha - i\beta, \quad (4)$$

where $|e_i\rangle, |e_k\rangle$ with $i \neq k$ are eigenstates of the drift Hamiltonian, \hat{H}_0 , $\alpha, \beta \in \mathbb{R}$ and at least one of them is nonzero. If (i, k) is a decoupled transition, the Lie algebra of the system contains the element

$$\hat{T} = \alpha \hat{F}_{i,k} + \beta \hat{G}_{i,k}, \quad (5)$$

and the generalized skew-Hermitian Pauli matrices $\hat{G}_{i,k}, \hat{F}_{i,k}$ and $\hat{D}_{i,k}$ are also elements of the Lie algebra $\text{Lie}(i\hat{H}_0, \hat{T})$, i.e. $\dim[\text{Lie}(i\hat{H}_0, \hat{T})] = 4$.

If a system contains the two decoupled transitions $\{(i, k), (i, l)\}$ with $\hat{T}_1 = \alpha_1 \hat{F}_{i,k} + \beta_1 \hat{G}_{i,k}$ and $\hat{T}_2 = \alpha_2 \hat{F}_{i,l} + \beta_2 \hat{G}_{i,l}$, respectively, and $i \neq l \neq k$, their commutator is of the form $\hat{T}_3 := [\hat{T}_1, \hat{T}_2] = \alpha_3 \hat{F}_{k,l} + \beta_3 \hat{G}_{k,l}$. The graph of the system thus has an additional edge (k, l) , as depicted in figure 1(a) for $i = 0, k = 1$ and $l = 2$. For testing controllability, it is often sufficient to know that there exists a transition (k, l) , without calculating the coefficients α_3, β_3 . Since the existence of the transition (k, l) can be deduced from the graph with edges (i, k) and (i, l) , we refer to this operation as graphical commutator and denote it as $(k, l) = [(i, k), (i, l)]$.

Graphical commutators can be used as an efficient tool to decouple resonant transitions. This is shown in the example depicted in figure 1(b). Consider transitions $\hat{T}_1 = \alpha_1 \hat{G}_{1,2} + \beta_1 \hat{F}_{1,2}$ (dash-dotted green arrow in figure 1(b)) and $\hat{T}_2 = \alpha_2 \hat{F}_{0,3} + \beta_2 \hat{G}_{0,3} + \gamma_2 \hat{F}_{1,4} + \delta_2 \hat{G}_{1,4}$ (dashed blue arrows) with \hat{T}_1 decoupled (i.e. the transition $(1, 2)$ is not coupled to any other one) and \hat{T}_2 consisting of the pair of coupled transitions $\{(0, 3), (1, 4)\}$, which are not coupled to any other transition. The graphical commutator between \hat{T}_1 and \hat{T}_2 has then only contributions corresponding to the transition $(2, 4)$, depicted by the red arrow. The graph has thus the additional decoupled transition $(2, 4)$. Furthermore, the graphical commutator between the decoupled transitions $(1, 2)$ and $(2, 4)$ is the decoupled transition $(1, 4)$. In this example, the resonant transitions $\{(0, 3), (1, 4)\}$ can thus be decoupled by taking graphical commutators.

In certain instances, the graphical commutators might not give enough information to decouple resonant transitions and determine the controllability of the system. Examples are shown in figures 1(c) and (d). In figure 1(c), the graphical commutator between the coupled transitions $\{(0, 1), (2, 3)\}$ (dashed blue arrows) and the decoupled transition $(1, 2)$ (dash-dotted green arrows) results in the coupled transitions $\{(0, 2), (1, 3)\}$ (red arrows). This does not allow for decoupling any of the coupled transitions. In figure 1(d), two pairs of coupled transitions are considered, namely $\{(0, 1), (2, 3)\}$ (dashed blue arrows) and $\{(0, 2), (1, 3)\}$ (dash-dotted green arrows). The graphical commutators $[(0, 1), (0, 2)]$ and $[(1, 3), (2, 3)]$ both result in the transition $(1, 2)$. Without further knowledge of the coefficients of the corresponding Lie algebra elements, it is not possible to determine whether the two terms result in a non-zero transition $(1, 2)$, or whether they cancel each other.

In such cases, we make use of an alternative procedure to decouple resonant transitions, which takes into account that, in qubit systems, the same (coupled) transitions are often driven by different controls. Consider a system with drift \hat{H}_0 and two controls that both drive the two resonant transitions $\{(1,2), (3,4)\}$. The Lie algebra of the system thus contains the terms

$$\begin{aligned}\hat{T}_4 &= \alpha_4 \hat{F}_{1,2} + \beta_4 \hat{G}_{1,2} + \gamma_4 \hat{G}_{3,4} + \delta_4 \hat{G}_{3,4}, \\ \hat{T}_5 &= \alpha_5 \hat{F}_{1,2} + \beta_5 \hat{G}_{1,2} + \gamma_5 \hat{G}_{3,4} + \delta_5 \hat{G}_{3,4}.\end{aligned}\quad (6)$$

The two transitions $(1,2)$ and $(3,4)$ are decoupled if $\hat{T}_6 := \alpha_6 \hat{F}_{1,2} + \beta_6 \hat{G}_{1,2}$ is an element of the Lie algebra $\text{Lie}(i\hat{H}_0, \hat{T}_4, \hat{T}_5)$ for any real α_6, β_6 (at least one of them nonzero). This is true if and only if $\dim [\text{Lie}(i\hat{H}_0, \hat{T}_4, \hat{T}_5)] = \dim [\text{Lie}(i\hat{H}_0, \hat{G}_{1,2}, \hat{G}_{3,4})] = 7$, i.e. when the generated sub-algebra has maximum dimension (for a given number of transitions). Note that we have deliberately chosen transitions $\{(1,2), (3,4)\}$, which have no vertex in common, i.e. they are disjoint. Restricting ourselves to disjoint transitions, the maximum dimension of the generated sub-algebra is $3n_t + 1$, where n_t is the number of transitions that are coupled. In order to determine, if a set of n_t transitions driven by n_t different controls is decoupled, it is thus sufficient to calculate the dimension of the sub-algebra with maximal dimension $3n_t + 1$. This is typically much smaller than the dimension of the Lie algebra of the complete system. If the transitions are not disjoint, the dimension of the relevant subalgebra scales quadratically with the number of transitions n_t . This, although feasible for a small number of transitions, would make the construction of the subalgebras more demanding.

In the following we use both methods, i.e. graphical commutators and the calculation of the dimension of small sub-algebras, in order to decouple resonant transitions. This allows us to extend the graph test for controllability to coupled subsystems with resonant transitions.

3.2. Algorithms for graph test of controllability for coupled subsystems with resonant transitions

Our graph test for controllability of a quantum system with resonant transitions is divided into several steps, depicted in figure 2. The main output of the complete algorithm is a variable conveying whether the system is controllable, not controllable or whether the test remains inconclusive.

As can be seen in figure 2, the first stage is creating the initial graph of the quantum system (A1). The step-by-step definition is found in algorithm 1. To compute this graph, we diagonalize the drift \hat{H}_0 , taking its eigenstates as the vertices of the graph. Then, we determine the nonzero transitions $\langle e_a | \hat{H}_j | e_b \rangle$ for all m controls and take these to be the edges (a, b) of the graph. Numerically, we take into account all transitions that are larger or equal than a certain tolerance, $\delta_H > 0$, i.e.

$$|\langle e_a | \hat{H}_j | e_b \rangle| \geq \delta_H \longrightarrow (a, b). \quad (7)$$

The next step is to determine which transitions are coupled. As defined in section 2, two or more transitions are coupled if and only if they have the same energy gap and are generated by the same control. Any transition not coupled to any other one is by definition a decoupled transition. Note that two transitions generated by different controls can never be coupled to one another. To numerically compare the energy gaps, we define a minimal tolerance $\delta_E > 0$ such that two transitions driven by the same control, (a, b) and (c, d) , are considered coupled if $|\Delta E_{a,b} - \Delta E_{c,d}| \leq \delta_E$.

We sort all the transitions of each control according to the energy gaps $\Delta E_{a,b}$ and separate them depending on whether they are coupled or not. If a coupled transition $(a, b)^{(j)}$ is generated by multiple controls \hat{H}_j , we also calculate and store the corresponding transition coefficients $\langle e_a | \hat{H}_j | e_b \rangle = \alpha_{a,b}^{(j)} + \beta_{a,b}^{(j)} i$. Note that here, we have introduced the notation $(a, b)^{(j)}$ to indicate that the transition (a, b) is driven by the control \hat{H}_j .

The output of algorithm 1 encompasses the *decoupled-graph*, the *coupled-transitions* and the *coupled-coefficients*. The first output, *decoupled-graph*, is a dictionary that contains all vertices (eigenstates) and all decoupled edges (decoupled transitions) of the systems graph. These are the elements that we can use directly to test the controllability of the system using theorem 1. In the second output, *coupled-transitions*, every set of coupled transitions is stored as a list of tuples (a, b) . Coupled transitions cannot be immediately used for the test in theorem 1, but are necessary for generating additional decoupled transitions, by using graphical commutators or subalgebras of resonant transitions. The last output variable *coupled-coefficients* consists of the entries $\langle e_a | \hat{H}_j | e_b \rangle = \alpha_{a,b}^{(j)} + \beta_{a,b}^{(j)} i$ for every coupled transition $(a, b)^{(j)}$ which are required for calculating the dimension of the subalgebras.

Once the initial graph of the system has been computed, we check if the graph containing only decoupled transitions (*decoupled-graph*) is connected. To do so, we determine the number of connected components in

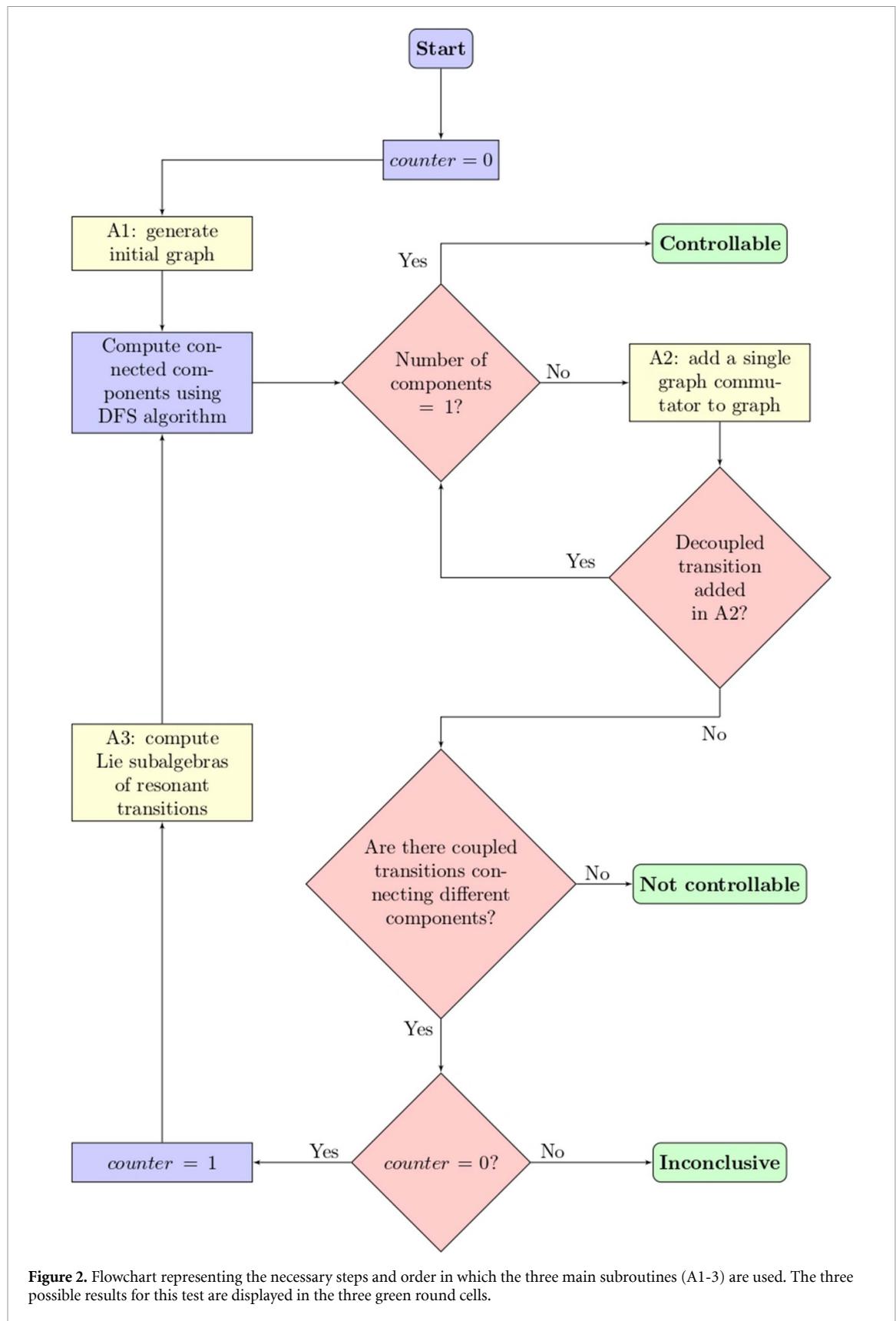


Figure 2. Flowchart representing the necessary steps and order in which the three main subroutines (A1-3) are used. The three possible results for this test are displayed in the three green round cells.

Algorithm 1. Compute the initial graph of the system.

```

1 Input:
  •  $H_{op}$ : list including the drift  $\hat{H}_0$  and all control operators  $\hat{H}_j$  in the logical basis.
  •  $\delta_H$ : minimum tolerance for a nonzero transition coefficient.
  •  $\delta_E$ : minimum tolerance for two energy gaps to be identical.
2 Diagonalize  $\hat{H}_0$  and represent all control operators  $\hat{H}_j$  in the eigenbasis of  $\hat{H}_0$ .
3 Initialize decoupled-graph, representing every eigenstate  $|e_a\rangle$  of  $\hat{H}_0$  as a vertex in the graph.
4 for every control  $\hat{H}_j$  in  $H_{op}$  do
5   for  $0 \leq a < b < \text{Hilbert space dimension}$  do
6     if  $|\langle e_a | \hat{H}_j | e_b \rangle| > \delta_H$  then
7       Add  $(a, b)$  to transition-list.
8     end if
9   end for
10  Sort transition-list by the energy gaps  $\Delta E_{a,b} := |E_a - E_b|$ 
11  for every  $(a, b)$  in transition-list do
12    Add all transitions  $(c, d)$  resonant to  $(a, b)$  (i.e.  $|\Delta E_{a,b} - \Delta E_{c,d}| < \delta_E$ ) to a new, separate list resonant-trans-ab (including  $(a, b)$ ).
13    Add coefficients  $\langle e_c | \hat{H}_j | e_d \rangle$  of all resonant transitions  $(c, d)$  into a new list resonant-coef-ab (including  $(a, b)$ ).
14    if resonant-trans-ab has more than one element then
15      Add resonant-trans-ab as a new entry in the list coupled-transitions.
16      Add resonant-coef-ab as a new entry in the list coupled-coefficients.
17    else
18      Add resonant-trans-ab as a decoupled transition in decoupled-graph.
19    end if
20    Remove all resonant transitions  $(c, d)$  and  $(a, b)$  from transition-list.
21  end for
22 end for
23 Output:
  • decoupled-graph: a dictionary containing all vertices of the graph (eigenstates of  $\hat{H}_0$ ) and all decoupled transitions as edges.
  • coupled-transitions: a list of lists containing all coupled transitions stored as dictionaries and sorted by their respective energy gap  $\Delta E_{a,b}$ . Resonant transitions from the same control with the same  $\Delta E_{a,b}$  are stored in the same inner list since they are coupled.
  • coupled-coefficients: a list of lists of the transition coefficients, matching the order of coupled-transitions.

```

the graph. In terms of graph theory, a component is a connected subgraph that it is not contained in any larger connected subgraph. In other words, each of the connected parts into which we can divide a graph is called a component of the graph. Therefore, the graph is connected if and only if it has exactly one component. One of the many possible ways to count the number of components is to use the depth-first search (DFS) algorithm². If the *decoupled-graph* is already connected, we can stop the routine and state that the system is indeed controllable.

If the graph is not connected, then algorithm 2 is called. The aim of this algorithm is to search for an additional edge of the graph by using graphical commutators. As explained in section 2, the graphical commutators of a coupled transition and a decoupled transition may generate a new decoupled transition. Note that, instead of identifying and adding every possible combination of commutators between a decoupled transition and a set of coupled transitions, we are only interested in commutators that result in a decoupled transition which connects different components of the graph and therefore potentially creates a connected graph. That is, in algorithm 2 we compute the commutator of a coupled transition that connects two separate components of *decoupled-graph* with a decoupled transition in one of the two components. If the result of this commutator is a decoupled transition, then this new edge will connect the two components in *decoupled-graph*, merging them into a single component. If a new decoupled transition is added to the *decoupled-graph* during this step, the number of connected components is counted again to determine if the graph is now connected.

If the number of components is still larger than one, i.e. if the *decoupled-graph* is not connected after algorithm 2, and no new transition was added during its last call, then we check if there is any transition in *coupled-transitions* that connects different components of *decoupled-graph*. If this is not the case, the routine is stopped. If all transitions, coupled and decoupled, only connect vertices within the same components, then

² The DFS algorithm is a common recursive algorithm to explore systematically all vertices in a graph. It uses an exhaustive search by traversing down a chain of edges, or by backtracking when not possible.

Algorithm 2. Add a single decoupled transition using graphical commutators.

```

1 Input:
  • decoupled-graph: from algorithm 1.
  • graph-components: list containing the sets representing the connected components of decoupled-graph.
  • coupled-transitions: from algorithm 1.
2 transition-added = False
3 connecting-transitions-found = False
4 i = 0
5 while (i < length of coupled-transitions) and (transition-added = False) do
6   | i = i + 1
7   | element_i = coupled-transitions[i]
    # element_i is a set of coupled transitions of the form  $\{(a_0, a_1), (a_2, a_3), \dots\}$ 
8   | j = 0
9   | while (j < length of element_i) and (transition-added = False) do
10    |   | j = j + 1
11    |   |  $(b_0, b_1) = element_i[j]$ 
12    |   | if  $b_0$  and  $b_1$  belong to different components in graph-components then
13    |   |   | connecting-transitions-found = True
14    |   |   | component_0 = component in graph-components containing  $b_0$ 
15    |   |   | component_1 = component in graph-components containing  $b_1$ 
16    |   |   | k = 0
17    |   |   | while (k < number of transitions in decoupled-graph) and (transition-added = False) do
18    |   |   |   | k = k + 1
19    |   |   |   |  $(c_0, c_1)$  = kth transition in decoupled-graph
20    |   |   |   | if  $\{c_0, c_1\} \subset component_0$  or  $\{c_0, c_1\} \subset component_1$  then
21    |   |   |   |   | graphical-commutator = [element_i,  $(b_0, b_1)$ ]
22    |   |   |   |   | if graphical-commutator is a single transition then
23    |   |   |   |   |   | transition-added = True
24    |   |   |   |   |   | Add graphical-commutator to decoupled-graph
25    |   |   |   |   |   | Merge component_0 and component_1 in graph-components
26    |   |   |   | end if
27    |   |   | end if
28    |   | end while
29    | end if
30 end while
31 end while
32 Output:
  • decoupled-graph: updated after the routine.
  • graph-components: updated after the routine.
  • transition-added: True if a transition was added during this subroutine to decoupled-graph; False otherwise.
  • connecting-transitions-found: True if at least one transition in coupled-transitions connects different components; False otherwise.

```

all possible commutators result in transitions within the same components. The number of components will stay constant and the graph will never be connected. This implies that the system is not controllable.

However, if there exists at least one coupled transition connecting different components, then this argument is no longer valid and the controllability of the system is not yet decided. For these cases, we use the *coupled-coefficients* that are related to the *coupled-transitions* to obtain a more conclusive answer.

This step is carried out by algorithm 3. It builds upon the updated *decoupled-graph* from algorithm 2 and tries to add new decoupled transitions by computing the low-dimensional subalgebras of resonant transitions driven by different controls. First, the algorithm selects the transitions from *coupled-transitions* that connect different components. In particular, we are exclusively interested in sets of disjoint transitions, such that the dimension of the associated subalgebras remains low. Next, the algorithm finds out if those transitions have a multiplicity equal to or higher than 2, that is, if the same sets of coupled transitions $\{(a_0, a_1), (a_2, a_3), \dots\}^{(j)}$ are generated multiple times by different controls \hat{H}_j . Following the reasoning presented in section 3.1, we try to generate the full associated subalgebra of the *coupled-transitions*, $\{(a_0, a_1), (a_2, a_3), \dots\}$, by using their respective *coupled-coefficients*. If a subalgebra with maximal dimension is generated, we take all transitions $\{(a_0, a_1), (a_2, a_3), \dots\}$ as effectively decoupled. This allows us to add every transition (a_k, a_l) as a decoupled edge to *decoupled-graph*. Note that in algorithm 3, we compute the Lie subalgebras of the resonant transitions including $i\hat{H}_0$, where we deduce the commutators

Algorithm 3. Subroutine to compute subalgebras of repeated resonant transitions.

```

1 Input:
  • decoupled-graph: from algorithm 2.
  • graph-components: from algorithm 2.
  • coupled-transitions: from algorithm 1.
  • coupled-coefficients: from algorithm 1.
2 for every set of resonant transitions  $\{(a_0, a_1), (a_2, a_3), \dots\}$  in coupled-transitions do
3   resonant-set =  $\{(a_0, a_1), (a_2, a_3), \dots\}$ 
4   if resonant-set contains only disjoint transitions then
5     if resonant-set connects different components in graph-components
      then
6       if resonant-set appears multiple times in coupled-transitions then
7         Set  $m$  as the number of times resonant-set is generated by different controls.
8         Set  $n_t$  as the number of coupled transitions in resonant-set
9         for  $0 \leq j < m$  do
10          Using the coefficients in coupled-coefficients related to the  $j$ th instance of resonant-set, generate an array
            T-array- $j$  to represent the transition.
            #  $\hat{T}_j = \alpha_{a_0, a_1}^{(j)} \hat{F}_{a_0, a_1} + \beta_{a_0, a_1}^{(j)} \hat{G}_{a_0, a_1} + \alpha_{a_2, a_3}^{(j)} \hat{F}_{a_2, a_3} + \beta_{a_2, a_3}^{(j)} \hat{G}_{a_2, a_3} + \dots$ 
            # T-array- $j$  =  $(\alpha_{a_0, a_1}^{(j)}, \beta_{a_0, a_1}^{(j)}, 0, \alpha_{a_2, a_3}^{(j)}, \beta_{a_2, a_3}^{(j)}, 0, \dots)$ 
11        end for
12        if  $\dim(\text{Lie}\{\hat{H}_0, \text{T-array-0}, \dots, \text{T-array-}m\}) = 3n_t + 1$  then
13          Add every transition in resonant-set to decoupled-graph
14        end if
15      end if
16    end if
17  end if
18 end for
19 Output:
  • decoupled-graph: updated after the routine.

```

$\left[i\hat{H}_0, \{(0, 1), (2, 3)\}^{(j)} \right]$ using equation (3). This makes these calculations simple and encompassed within a vector space of dimension $3n_t$, with n_t being the number of transitions in $\{(a_0, a_1), (a_2, a_3), \dots\}$.

Finally, we compute the new number of components in *decoupled-graph* using DFS once again. If the graph is still not connected, we use algorithm 2 to add as many graphical commutators as possible to connect the remaining components. If the graph ends up with one single component, the system is controllable. If there are more components and there are no transitions in *coupled-transitions* that connect any of them, then the system is proven to be not controllable. If none of these two hold to be true, then the controllability test remains inconclusive. Indeed, unlike the dynamical Lie algebra method, the test might yield no definitive answer. If a graph is not connected, we cannot ensure that the system is not controllable, generally speaking. It might be that the graph can be connected after computing a large number of commutators or that it will remain forever not connected. Calculating this quantity of commutators is tantamount to computing the dynamical Lie algebra and would therefore be unfeasible for relatively large systems.

3.3. Illustrative examples

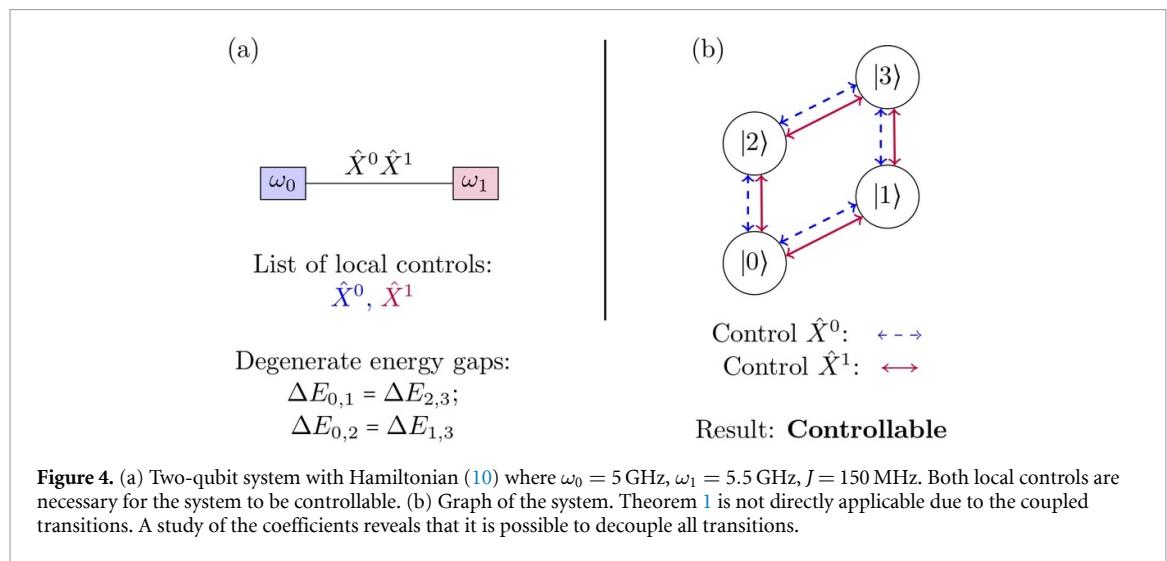
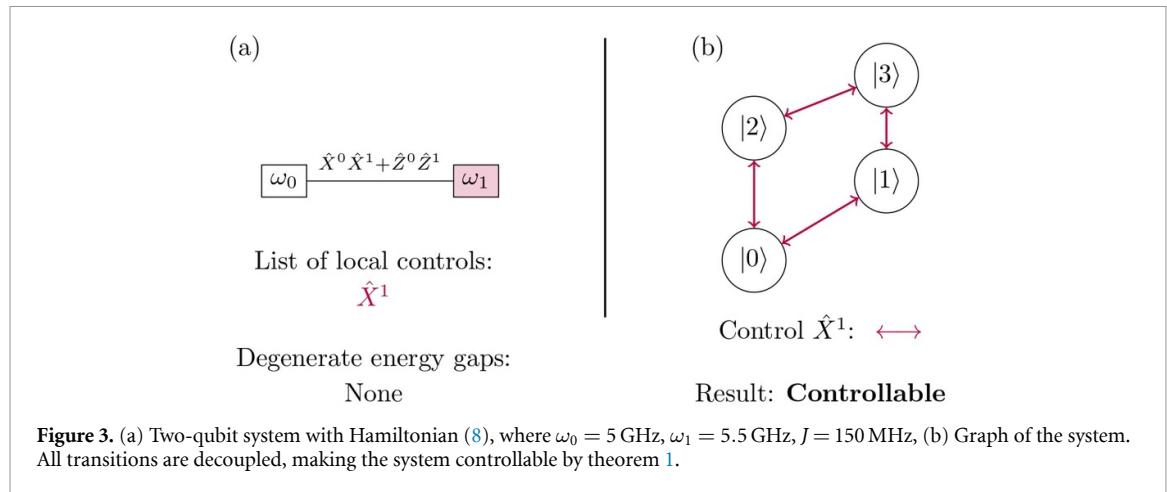
Here we present two examples constructed to showcase how the algorithms described in section 3.2 work. We start by considering the two-qubit system shown in figure 3(a), described by the Hamiltonian

$$\hat{H}_{2A}(t) = \hat{H}_0 + u_1(t)\hat{H}_1 \quad (8)$$

with the drift

$$\hat{H}_0 = \sum_{i=0}^1 -\frac{\omega_i}{2} \hat{Z}^i + J(\hat{X}^0 \hat{X}^1 + \hat{Z}^0 \hat{Z}^1), \quad (9)$$

and a single local control $\hat{H}_1 = \hat{X}^1$ acting on qubit 1. Here, ω_i are the natural frequencies of the qubits and J is the coupling strength. For simplicity, we use the notation $\hat{X}^j = \hat{\sigma}_x^j$ for the Pauli matrices. Algorithm 1 results in the graph shown in figure 3(b) with the circular vertices representing the eigenstates of \hat{H}_0 . The local control \hat{H}_1 gives rise to four non-zero transitions, which form the edges of the graph, denoted by the arrows in figure 3(b). The different lengths of the arrows indicate that all four transitions have different energy gaps and are thus decoupled. The actual controllability test for this example consists simply of



algorithm 1 and the DFS algorithm, which proves that the graph is connected and the system is controllable. In other words, from every vertex $i = 0, \dots, 3$, every other vertex of the graph can be reached by following the edges belonging to decoupled transitions, as seen in figure 3(b).

A second example is illustrated in figure 4(a). Here, the Hamiltonian is given by

$$\hat{H}_{2B}(t) = \hat{H}_0 + \sum_{j=1}^2 u_j(t) \hat{H}_j \quad (10)$$

with the drift

$$\hat{H}_0 = \sum_{i=0}^1 -\frac{\omega_i}{2} \hat{Z}^i + J \hat{X}^0 \hat{X}^1, \quad (11)$$

and two local controls

$$\hat{H}_1 = \hat{X}^1 \quad \text{and} \quad \hat{H}_2 = \hat{X}^0 \quad (12)$$

acting on the qubits 1 and 0, respectively. The main difference between equations (8) and (10) is the omission of $\hat{Z}^0 \hat{Z}^1$ in the drift Hamiltonian. This severely impacts the graph, as seen in figure 4. The symmetries that had previously been broken by including $\hat{Z}^0 \hat{Z}^1$ appear in the energy gap degeneracies, $\Delta E_{0,1} = \Delta E_{2,3}$ and $\Delta E_{0,2} = \Delta E_{1,3}$. There are no decoupled transitions and the number of components in the initial graph is equal to four. A second difference is the second local control. The transitions driven by the controls \hat{X}^1 (which is the same local control as in equation (8)) and \hat{X}^0 consist of two pairs of coupled transitions $\{(0,1), (2,3)\}$ and $\{(0,2), (1,3)\}$ shown by purple solid and dashed blue arrows respectively in figure 4(b).

Table 1. Control coefficients $\langle e_a | \hat{H}_j | e_b \rangle$ for the system (10). The coupled transitions are $\{(0, 1), (2, 3)\}$ and $\{(0, 2), (1, 3)\}$. They appear with double multiplicity, once for each control.

	Transition parameters			
	(0, 1)	(2, 3)	(0, 2)	(1, 3)
\hat{X}^0	0.967	0.967	-0.253	0.253
\hat{X}^1	-0.281	0.281	-0.960	-0.960

In this example, graphical commutators do not yield any new transition that we could add to the graph. In fact, we can only take commutators of pairs of coupled transitions, analogously to the case depicted in figure 1(d). Since there is no definitive answer, graphical commutators are not useful in this particular instance. However, since there exist coupled transitions which connect different components of the graph, algorithm 3 can be applied. The coefficients $\langle e_a | \hat{H}_j | e_b \rangle = \alpha_{a,b}^{(j)} + \beta_{a,b}^{(j)} i$ of the coupled transitions are shown in table 1. Note that in this example, the coefficients are real, $\beta_{a,b}^{(j)} = 0$. The Lie algebra thus contains the terms

$$\begin{aligned}\hat{T}_0 &= \alpha_{0,1}^{(0)} \hat{F}_{0,1} + \alpha_{2,3}^{(0)} \hat{F}_{2,3}, \\ \hat{T}_1 &= \alpha_{0,1}^{(1)} \hat{F}_{0,1} + \alpha_{2,3}^{(1)} \hat{F}_{2,3}.\end{aligned}\quad (13)$$

It is immediate to see that we can isolate the term $\hat{F}_{2,3}$ by a linear combination of \hat{T}_0 and \hat{T}_1 . This means that all operators $\hat{F}_{2,3}, \hat{G}_{2,3}, \hat{D}_{2,3}$ are elements of the subalgebra. Similarly, the opposite linear combination can isolate the element $\hat{F}_{0,1}$, adding $\hat{F}_{0,1}, \hat{G}_{0,1}, \hat{D}_{0,1}$ to the generated subalgebra. This means that the dimension of $\text{Lie}(i\hat{H}_0, \hat{T}_0, \hat{T}_1) = 7$ (including the contribution of $i\hat{H}_0$) is maximal and thus the transitions $(0, 1)$ and $(2, 3)$ can be decoupled. Analogously, $(0, 2)$ and $(1, 3)$ may be decoupled as well. This means that all the edges are decoupled in the graph of figure 4. By virtue of theorem 1, this implies controllability of the system.

4. Results

In this section, we examine qubit systems which are used for quantum computing. With the algorithm presented in section 3, we can efficiently prove if a given qubit array is controllable. We present three different examples based on the IBM five-qubit array *ibmq_quito* [24]. We do not use the exact parameters of this system, but simply its configuration and realistic parameters (except for the natural frequencies which correspond to the real data [24]).

The Hamiltonian of an array similar in structure to *ibmq_quito* can be expressed as:

$$\hat{H}^{\text{quito}}(t) = - \sum_{j=0}^4 \frac{\omega_j}{2} \hat{Z}^j + \hat{H}_c^{\text{quito}} + \sum_{k=1}^m u_k(t) \hat{H}_k, \quad (14)$$

where $u_k(t)$ represent local controls and the couplings \hat{H}_c^{quito} are of the form

$$\hat{H}_c^{\text{quito}} = J_{0,1} \hat{H}_{0,1} + J_{1,2} \hat{H}_{1,2} + J_{1,3} \hat{H}_{1,3} + J_{3,4} \hat{H}_{3,4} \quad (15)$$

with each $\hat{H}_{j,k}$ representing an entangling (time-independent) coupling between the qubits j and k . The following three examples are based on equation (14) for different types of couplings and local controls. We will prove that while the original IBM design is indeed controllable there exist options that require fewer resources for controllability.

The set of parameters, including the natural frequencies ω_j and the coupling strengths $J_{i,j}$ are found in table 2. We use these parameters for all examples presented in this section.

4.1. Example A: five-qubit system similar to IBM's *ibmq_quito*

This example mimics the qubit arrangement of *ibmq_quito*. The static couplings in equation (15) are [25]

$$\hat{H}_{i,j} = J_{i,j} (\hat{X}^i \hat{X}^j + \hat{Y}^i \hat{Y}^j). \quad (16)$$

A local control $\hat{H}_j = \hat{X}^j$ is added to every qubit. The system diagram is shown in figure 5.

For a five-qubit system, there is a total of 32 vertices. After running algorithm 2 for this example with an energy gap tolerance of $\delta_E = 0.01$ GHz, we obtain an initial graph with only 7 decoupled edges, resulting in a total of 25 different connected components. Use of algorithm 2 alone is not enough in this case, since the

Table 2. Parameters for the system (14). The qubit frequencies are those of [24]. The coupling strengths have been chosen to be in the range of hundreds of MHz, common for this type of qubits.

Coupling strengths (MHz)				
$J_{0,1}$	$J_{1,2}$	$J_{1,3}$	$J_{3,4}$	
100	250	170	300	
Qubit frequencies (GHz)				
ω_0 5.301	ω_1 5.081	ω_2 5.322	ω_3 5.164	ω_4 5.052

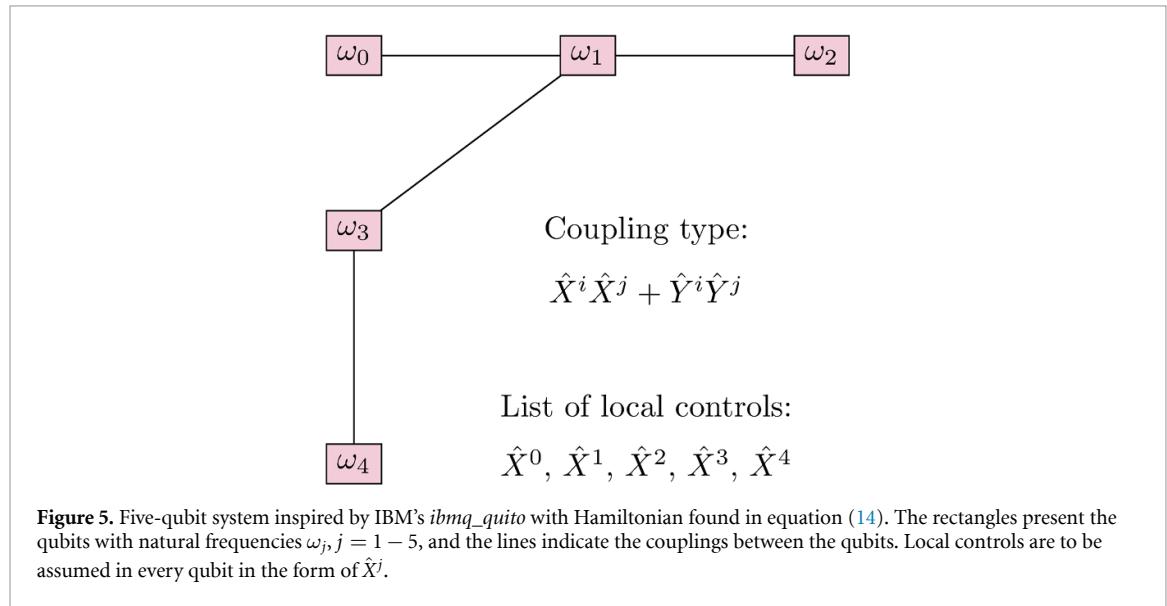


Figure 5. Five-qubit system inspired by IBM's *ibmq_quito* with Hamiltonian found in equation (14). The rectangles present the qubits with natural frequencies ω_j , $j = 1 - 5$, and the lines indicate the couplings between the qubits. Local controls are to be assumed in every qubit in the form of \hat{X}^j .

graphical commutators do not yield any new results. Executing algorithm 3, a total of 192 new edges are added to the graph. These turn out to be more than sufficient to connect all the components and achieve a connected graph.

We can thus conclude that the system presented in figure 5 is controllable and therefore suitable to perform any unitary operation. In the following examples we investigate whether the five-qubit array is also controllable if the number of local controls is reduced.

4.2. Example B: five-qubit system with reduced number of controls

Next, we investigate the controllability of a five-qubit system with a reduced number of local controls. For the drift Hamiltonian, we consider the same five-qubit system as before with all parameters as listed in table 2 but decrease the number of local controls to two. This is depicted in figure 6, which represents one of the ten different scenarios with two local controls $\hat{H}_j = \hat{X}^j$.

Running algorithm 1 we obtain identical results to those in section 4.1: a graph with only seven decoupled transitions and 25 different connected components. Again, it is crucial to use algorithm 3 to make the graph connected. It may come as a surprise that the number of new edges that we can add is again 192. In other words, the number of new decoupled edges has not been reduced by removing three of the local controls. With the new decoupled edges included, the system is indeed controllable.

We could attempt to take this study further to single local control. However, with a single local control \hat{X}^j , we always obtain an inconclusive result, no matter on which qubit we apply it. Although this hints to the possibility of the system not being controllable, the algorithm cannot conclusively determine it, similarly to the case shown in figure 4. In that two-qubit example we can prove using the Lie algebra method that a single local control \hat{X}^j is never sufficient. We therefore conjecture that the least number of local controls required for the system to be controllable is two. One may wonder whether independent \hat{X}^j - and \hat{Y}^j -controls acting on the same qubit change the controllability. Assuming the local, field-free Hamiltonians to be proportional to \hat{Z}^j , there is no additional benefit from a second control, since a single local control \hat{X}^j that does not commute with \hat{H}_0 is sufficient to also generate a \hat{Y}^j contribution and viceversa.

Since two local controls is the minimum number to obtain a controllable system, we now study how the position of the control affects controllability. We categorize all possible arrangements of two local controls \hat{X}^j

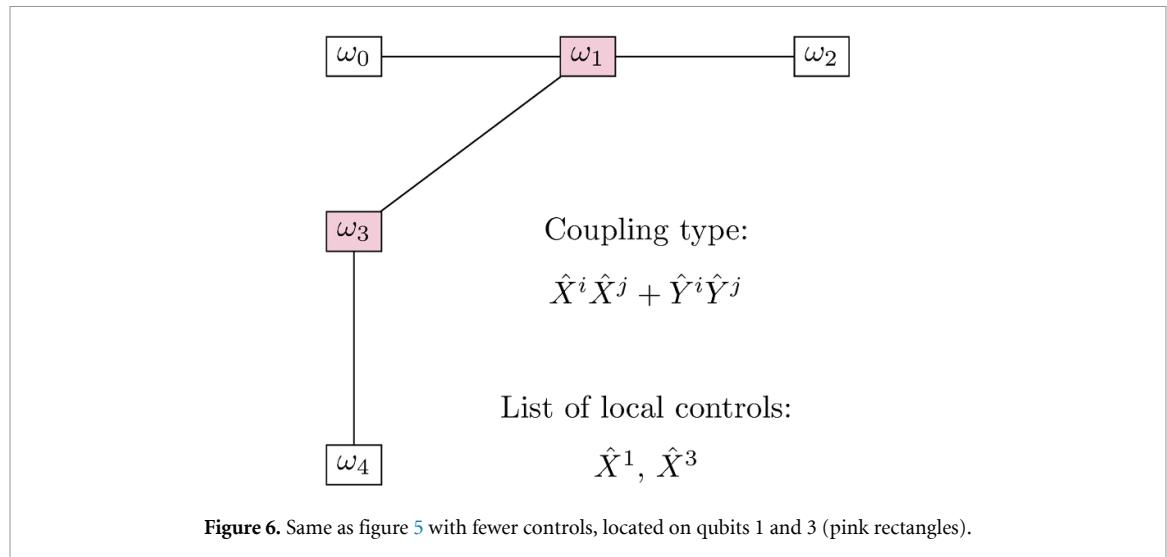


Table 3. Study of controllability for a five-qubit system similar to *ibmq_quito* but with only two local controls.

Distance	Controllability	# cases
1	Controllable	4
2	Inconclusive	4
3	Controllable	2

by their distance. For example, the local controls for qubits 1 and 3, shown in figure 6, have distance equal to 1 since they are direct neighbors connected by a single coupling. Analogously, we can set local controls separated by two couplings (e.g. on qubit 2 and 3) that have distance 2, or local controls for qubits separated by three couplings (e.g. qubits 0 and 4) that have distance 3. The results of the controllability tests for all these possible arrangements of two local controls are shown in table 3.

Accordingly, the five-qubit system is controllable with two local controls if they have either distance one or three, independent of the position of the controls themselves. On the other hand, if the controls are at distance 2, the algorithm returns an inconclusive answer. This implies that algorithm 3 cannot decouple enough transitions for the graph to be connected. In fact, it turns out that no new transitions could be added at all. This is because the coefficients $\langle e_a | \hat{H}_j | e_b \rangle$ of the different transitions depend on the position of the local control that drives them. By shifting a local control to a neighboring qubit, it is common to see a change in some of the coefficients. The transition coefficients of same-type controls for a set of coupled transitions tend to be linearly independent when the controls are shifted by an odd number of qubits and dependent in the even case. This allows us to generate maximal subalgebras only for controls that are at an odd qubit-distance, turning the coupled transitions into decoupled transitions that can eventually make the graph connected.

4.3. Example C: five-qubit system with single local control

To conclude the list of examples, we demonstrate that the number of local controls that is required for the system to be controllable also depends on the type for couplings connecting the qubits. We showcase a system that is controllable with a minimal number of local controls, i.e. a single local control. The main difference between this example (shown in figure 7) and the previous ones (displayed in figures 5 and 6) is that the couplings are of the form

$$\hat{H}_{i,j} = J_{i,j} \left(\hat{X}^i \hat{X}^j + \hat{Y}^i \hat{Y}^j + \hat{Z}^i \hat{Z}^j \right), \quad (17)$$

i.e. they have an extra term $\hat{Z}^i \hat{Z}^j$ compared to the couplings considered in the examples before. These kinds of couplings can be achieved in systems of superconducting qubits by harnessing both capacitive and inductive interactions between the qubits. Executing algorithm 1 with $\delta_E = 0.1$ GHz, we obtain a total of 24 initially decoupled edges and a total of 14 connected components. This is already more promising than the examples in sections 4.1 and 4.2. Furthermore, use of graphical commutators results in the system being controllable, ending up with 37 decoupled edges that are sufficient to make the graph connected. The controllability analysis thus shows that with the $\hat{Z}^i \hat{Z}^j$ addition, the system is controllable with only one local control. In figure 7, we have added a local control on qubit 1, but additional tests show that the same statement is true

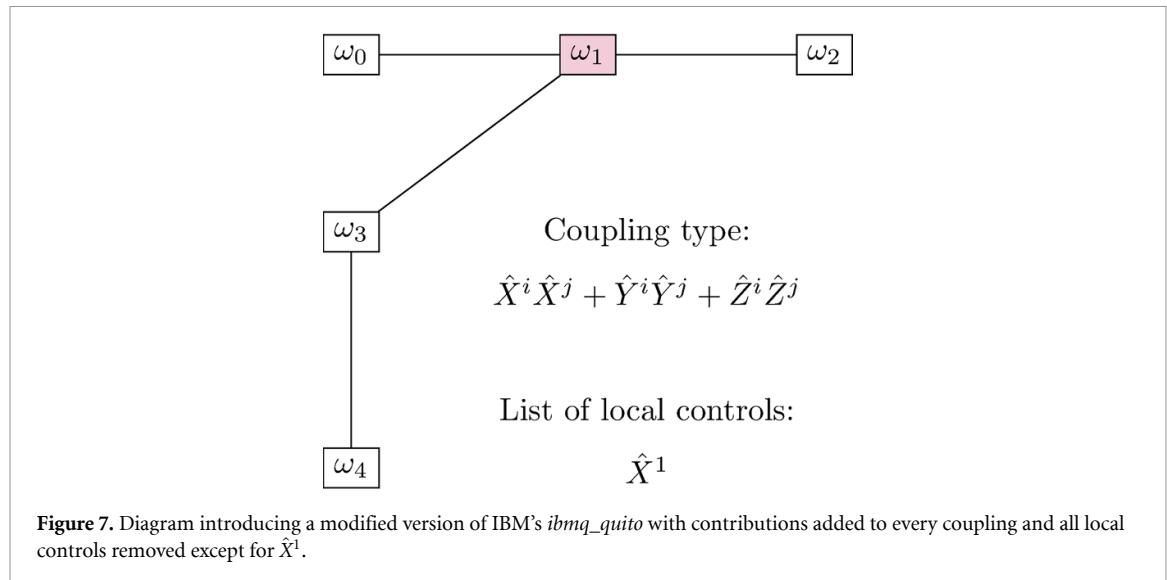


Figure 7. Diagram introducing a modified version of IBM's *ibmq_quito* with contributions added to every coupling and all local controls removed except for \hat{X}^1 .

for any single local control \hat{X}^j , independent of its position at qubit 0 to 4. Therefore, this controllable system has a minimal number of local controls at the cost of using a more complex type of two-qubit couplings.

The effect of couplings of the form $\hat{Z}^i \hat{Z}^j$ on the graph of the system can be understood by considering the simple two-qubit systems shown in figures 3 and 4. In figure 4, couplings of the form $\hat{X}^i \hat{X}^j$ result in two pairs of resonant (coupled) transitions for each local control. Therefore, two local controls are required to decouple the transitions and ensure controllability. In figure 3, a single control \hat{X} is sufficient due to the $\hat{Z}^0 \hat{Z}^1$ term in its coupling. This term is already diagonal in the eigenbasis of the free qubits, which means that it does not mix the eigenstates of the free qubits. However, it shifts their eigenenergies, such that all four vertices of the graph correspond to different transition frequencies and thus a single local control corresponds to four decoupled transitions, making the system controllable. A similar effect happens in the present five-qubit example, where the couplings have an off-diagonal contribution $\hat{X}^i \hat{X}^j + \hat{Y}^i \hat{Y}^j$ and a diagonal part $\hat{Z}^i \hat{Z}^j$. The former mixes the free-qubit eigenstates and in turn creates more edges in the graph with the inclusion of a local control like \hat{X}^j . The latter changes some resonant energy gaps and decouples the transitions associated to them. The combination of these two effects achieves controllability with a minimal number of local controls, as seen in figure 7. The special role of this coupling has already been observed for spin systems, where spin chains have been proven to be controllable with a number of local controls smaller than the number of spins [11].

5. Summary and conclusions

We have presented a practical way to test for evolution-operator controllability in order to numerically verify the ability of a given qubit array to implement universal quantum computing. The graph theory-based approach allows for analyzing comparatively large qubit systems for which the evaluation of the Lie rank condition for controllability would be difficult or impossible. Indeed, the computationally expensive calculation of nested commutators of the drift and control Hamiltonians, which is required to construct the system's dynamical Lie algebra, is avoided in graphical methods. On the other hand, in some cases, a graphical controllability test remains inconclusive. We believe that the disadvantage is small in view of the positive results that can be obtained for systems not amenable to the Lie rank condition.

The key challenge in controllability tests for systems of coupled qubits using graph theory-based methods is due to the tensor product structure of Hilbert space. This leads to graphs that often consist of sets of coupled (resonant) transitions. A similar problem was encountered in the controllability analysis of driven quantum rotors [10] which are highly degenerate systems with multiple resonant transitions. In that case, the specific spectral structure allowed for an inductive evaluation of the graphical commutators [19]. Here, we have made no assumptions on the spectral structure of the system. Instead, we have found and implemented numerically efficient ways to decouple the resonant transitions such as to make them exploitable for the graph test. In particular, our algorithm only determines those graphical commutators which are relevant because they add to the connectivity of the graph. Moreover, we have made use of fact that in qubit arrays, coupled transitions are typically driven by several controls: These coupled transitions become decoupled if the dimension of the subalgebra that they generate when driven by different controls is maximal. Since the

sub-algebras are very small compared to the Hilbert space of the complete system, their numerical construction does not pose a challenge. The remaining limitation to the number of qubits N_Q , for which a controllability test can be carried out, is the exponential scaling of the Hilbert space dimension with N_Q because the drift Hamiltonian (including the qubit–qubit couplings) needs to be diagonalized. Typically, the drift is sparse in the logical basis. For the graph test presented here, only a single diagonalization of the (sparse) drift Hamiltonian is necessary to obtain the initial graph.

We have illustrated the utility of our approach by showing how controllability analysis can be used to determine the minimum number of local controls required for universal quantum computing in existing quantum processing units. To this end, we have chosen the five-qubit array *ibmq_quito* [24] as specific example, treating the qubit–qubit couplings as fixed. This is justified since, whenever a system with time-independent couplings is controllable, the corresponding system with tunable couplings is controllable as well. The actual *ibmq_quito* system contains a local control for each of the five qubits. Our analysis has shown that by modifying the couplings between the qubits, the number of local controls necessary for universal quantum computing can be reduced to a single local control. For standard qubit–qubit couplings the minimal number of local controls is two. Analyzing the controllability for different positions of the two local controls reveals that the distance between the local controls is essential for the system to be controllable: the two local controls must be separated by an odd number (one or three) of qubit–qubit couplings in order to provide controllability and allow for universal quantum computing. These results showcase the utility of our graph test for improving the design and scalability of qubit arrays.

In future work, it will be interesting to gather a deeper understanding of the required control distance, as this might allow for generalizing our findings to larger qubit arrays. One natural next step will be extending our approach from local to non-local controls, as obtained when using e.g. tunable couplers. Our graph test is limited by the exponential scaling of the Hilbert space, a common issue among algorithms run on a classical computer, which can only be overcome by mapping the controllability analysis to a quantum device. Another important open question is the relationship between the minimal number of local controls and the quantum speed limit for universal quantum computing. The quantum speed limit [26] refers to the minimum time in which a quantum process, such as a quantum gate, can be executed. For a complete universal set of gates, the quantum speed limit can be obtained using quantum optimal control [27]. Given a quantum system and a set of controls, the time required for any particular operation will always be equal or longer than the time required for the same operation in the same system with more controls added. Conversely, reducing the number of local controls is likely to increase the time needed to carry out a gate. Thus, reducing the number of local controls has to be balanced with the requirement of sufficiently fast logical operations. The minimal sets of local controls from controllability analysis as suggested here offers a good starting point to find the best balanced set of controls with quantum optimal control. Finally, a further important perspective for future work is the question of controllability of systems coupled to the environment. It will be interesting to see whether graph methods can be utilized in this context.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

We would like to thank David Pohl, Max Werninghaus, Eugenio Pozzoli, Daniel Basilewitsch, Ugo Boscain, and Mario Sigalotti for the helpful discussions. We gratefully acknowledge financial support from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 765267 (QuSCo) and the Einstein Research Foundation (Einstein Research Unit on Near-Term Quantum Devices).

ORCID iDs

Fernando Gago-Encinas  <https://orcid.org/0000-0001-6689-6346>
Monika Leibscher  <https://orcid.org/0000-0001-6006-9067>
Christiane P Koch  <https://orcid.org/0000-0001-6285-5766>

References

- [1] Glaser S J *et al* 2015 *Eur. Phys. J. D* **69** 279
- [2] Koch C P *et al* 2022 *EPJ Quantum Technol.* **9** 19
- [3] d’Alessandro D 2008 *Introduction to Quantum Control and Dynamics* (Chapman and Hall/CRC))

- [4] Schirmer S G, Fu H and Solomon A I 2001 *Phys. Rev. A* **63** 063410
- [5] Fu H, Schirmer S G and Solomon A I 2001 *J. Phys. A: Math. Gen.* **34** 1679
- [6] Altafini C 2002 *J. Math. Phys.* **43** 2051–62
- [7] Chambrion T, Mason P, Sigalotti M and Boscain U 2009 *Ann. Inst. Henri Poincare C* **26** 329–49
- [8] Boussaïd N, Caponigro M and Chambrion T 2013 *IEEE Trans. Autom. Control* **58** 2205–16
- [9] Boscain U, Caponigro M, Chambrion T and Sigalotti M 2012 *Commun. Math. Phys.* **311** 423–55
- [10] Leibscher M, Pozzoli E, Pérez C, Schnell M, Sigalotti M, Boscain U and Koch C P 2022 *Commun. Phys.* **5** 1–16
- [11] Wang X, Pemberton-Ross P and Schirmer S G 2012 *IEEE Trans. Autom. Control* **57** 1945–56
- [12] Wang X, Burgarth D and Schirmer S 2016 *Phys. Rev. A* **94** 052319
- [13] Chen J, Zhou Y, Bian J, Li J and Peng X 2020 *Phys. Rev. A* **102** 032602
- [14] Albertini F and D'Alessandro D 2021 *Syst. Control Lett.* **151** 104913
- [15] Albertini F and D'Alessandro D 2002 *Linear Algebr. Appl.* **350** 213–35
- [16] Boscain U, Caponigro M and Sigalotti M 2014 *J. Differ. Equ.* **256** 3524–51
- [17] Godsil C and Severini S 2010 *Phys. Rev. A* **81** 052316
- [18] Gokler C, Lloyd S, Shor P and Thompson K 2017 *Phys. Rev. Lett.* **118** 260501
- [19] Pozzoli E, Leibscher M, Sigalotti M, Boscain U and Koch C P 2022 *J. Phys. A: Math. Theor.* **55** 215301
- [20] Figgatt C, Ostrander A, Linke N M, Landsman K A, Zhu D, Maslov D and Monroe C 2019 *Nature* **572** 368–72
- [21] Krantz P, Kjaergaard M, Yan F, Orlando T P, Gustavsson S and Oliver W D 2019 *Appl. Phys. Rev.* **6** 021318
- [22] Schirmer S, Pullen I and Solomon A 2002 *J. Phys. A: Math. Gen.* **35** 2327
- [23] Boscain U, Pozzoli E and Sigalotti M 2021 *SIAM J. Control Optim.* **59** 156–84
- [24] IBM quantum (available at: <https://quantum-computing.ibm.com/services>) (Accessed 16 January 2023)
- [25] Roth M, Ganzhorn M, Moll N, Filipp S, Salis G and Schmidt S 2017 *Phys. Rev. A* **96** 062323
- [26] Deffner S and Campbell S 2017 *J. Phys. A: Math. Theor.* **50** 453001
- [27] Goerz M H, Motzoi F, Whaley K B and Koch C P 2017 *npj Quantum Inf.* **3** 37