Journal of Applied Physics

RESEARCH ARTICLE | MARCH 10 2025

Realizing permutation gates with phi-bits: Acoustic quantum analogue computing

David Cavalluzzi ; Akinsanmi S. Ige ; Keith Runge ; Pierre A. Deymier

J. Appl. Phys. 137, 104901 (2025)
https://doi.org/10.1063/5.0241680
CHORUS

Journal of Applied Physics ARTICLE pubs.aip.org/aip/jap

Realizing permutation gates with phi-bits: Acoustic quantum analogue computing

Cite as: J. Appl. Phys. **137**, 104901 (2025); doi: 10.1063/5.0241680
Submitted: 1 October 2024 · Accepted: 13 February 2025 ·
Published Online: 10 March 2025

David Cavalluzzi,[1,2] Akinsanmi S. Ige,[2,3,a)] Keith Runge,[2,3] and Pierre A. Deymier[2,3]

AFFILIATIONS

[1]Department of Electrical and Computer Engineering, The University of Arizona, Tucson, Arizona 85721, USA
[2]New Frontiers of Sound Science and Technology Center, The University of Arizona, Tucson, Arizona 85721, USA
[3]Department of Materials Science and Engineering, The University of Arizona, Tucson, Arizona 85721, USA

[a)]**Author to whom correspondence should be addressed:** akinsanmiige@arizona.edu

ABSTRACT

We present both the theoretical framework and experimental implementation of permutation gates using logical phi-bits, classical acoustic analogs of qubits. Logical phi-bits are nonlinear acoustic modes supported by externally driven acoustic metamaterials. Using a tensor product of modified Bloch sphere representations, we realize all possible two logical phi-bit permutations including SWAP and C-NOT. We also illustrate the scalability of a permutation for any number of logical phi-bits. Experimental demonstrations of these permutations require a single physical action on the driving conditions of the acoustic metamaterial. All logical phi-bits exist in the same physical system. We compare the phi-bit system with its quantum counterpart using Qiskit simulations, which illustrate the complexity of realizing these permutations in a quantum context.

I. INTRODUCTION

Permutations are key elements in many types of algorithms with application to cryptography, optimization, statistics, and analysis.[1–6] For digital computers, permutations pose a significant challenge in terms of efficiency and computing time when operating on large datasets. The parallelism of quantum computers holds the promise of greater efficiency and reduced time. However, increasing the number of qubits challenges state-of-the-art quantum computing platforms due to system decoherence leading to the accumulation of errors.[1] Quantum algorithms resort to performing circuits of single and double qubit gates for realizing permutations.[7,8]

Quantum computing is essentially phase computing where a coherent superposition of state of N qubit is defined by a multidimensional vector with 2^N complex components living in an associated Hilbert space. The basis of the Hilbert space depends on the choice of the representation of the multi-qubit states. Quantum correlations ought to make these components dependent on each other and manipulable in a massively parallel manner. A computation would then involve acting on the system to change all components of the complex vector at once in a controllable manner through a unitary operation (or gate), i.e., a rotation of the vector in the Hilbert space. One of the challenges in quantum computing is the fragility of multi-qubit superpositions arising from quantum decoherence of the wavefunction due to perturbations such as thermal fluctuations.[9] Another challenge is the weak correlation resulting from computing hardware constraints of limiting the number of qubits that can be manipulated simultaneously.[10] Recently, mechanical systems have shown great potential for performing computations. Such systems include acoustic metasurfaces,[11–14] conventional phononic crystals,[14] topological phononic crystals,[15,16] and granular materials.[17] In prior work, we demonstrated that nonlinear acoustic metamaterials can support logical phi-bits (acoustic qubit analogs), in which superposition of states can also be represented in an exponentially scaling Hilbert space to achieve scalable gates for any number of phi-bits.[18,19] A logical phi-bit is a nonlinear acoustic mode supported by an externally driven acoustic metamaterial, which can exist in robust superpositions of states that are not subjected to wavefunction decoherence. The two-level logical phi-bit is a classical analog of a qubit. Logical phi-bits coexist in the same physical domain defined by the metamaterial and nonlinearity provide the strong correlation necessary to realize massively parallel operations on multi-phi-bit states without hardware constraints.[18]

In this work, we present both the theoretical and experimental realization of permutations using a system of logical phi-bits. In Sec. II of this study, we define logical phi-bit as a nonlinear acoustic mode supported by a metamaterial composed of linearly arranged acoustic waveguides.[18] We introduce a tensor product of modified Bloch sphere representations of multi-phi-bit states, which enables us to show in Sec. III that we can realize all possible two logical phi-bit permutations including SWAP and C-NOT. We also show experimentally that a single physical action on the driving condition of the acoustic metamaterial is sufficient to perform these permutations. In Sec. IV, we demonstrate the scalability of a permutation for any number of logical phi-bits. Section V provides a comparison of the phi-bit system with its quantum counterpart using Qiskit simulations. In Sec. VI, we draw conclusions about the advantages of logical phi-bit permutations.

II. PHYSICAL SYSTEM AND LOGICAL PHI-BIT

Logical phi-bits are generated with an acoustic metastructure consisting of three aluminum waveguides, 60 cm long (approximately 1 cm in diameter) arranged linearly in a parallel array with a lateral spacing of 2 mm filled with epoxy.[18] The waveguides are labeled 1, 2, and 3, with waveguide 2 positioned between 1 and 3. At one end of waveguides 1 and 2, sinusoidal waves of equal amplitude 80 V are generated with frequency $f_1 = 62\,\text{kHz}$ and $f_2 = 66\,\text{kHz}$, respectively, using ultrasonic transducers. Each transducer is driven by two separate signal generators to reduce unwanted crosstalk. The third waveguide remains unexcited. The amplitude and the phase of the propagated wave are measured at the other end of the three waveguides using another set of ultrasonic transducers connected to an oscilloscope. The experimental setup is shown and illustrated schematically in Fig. 1.

The measured amplitudes were transformed to the frequency domain using Fast Fourier Transform (FFT). On the spectral domain, we observe strong peaks at the primary frequencies, f_1 and f_2, as well as secondary peaks corresponding to nonlinear modes at frequencies given by linear combinations of the primary frequencies,

$$F^{(p,q)} = pf_1 + qf_2. \tag{1}$$

In Eq. (1), p and q are the integer coefficients of the linear combination of the primary frequencies. The modes corresponding to the secondary peaks' frequencies $F^{(p,q)}$ are identified as the logical phi-bits. Therefore, a logical phi-bit is said to exist in the spectral domain of this setup and all phi-bits, i.e., nonlinear modes coexist in the same physical domain.

In the spectral domain, the complex amplitude of a nonlinear mode with frequency $p\omega_1 + q\omega_2$ at the end of the three waveguides can be expressed as

$$U^{(p,q)} = \begin{pmatrix} C_1 e^{i\phi_1^{(p,q)}} \\ C_2 e^{i\phi_2^{(p,q)}} \\ C_3 e^{i\phi_3^{(p,q)}} \end{pmatrix} e^{i(p\omega_1 + q\omega_2)t}, \tag{2}$$

where C_1, C_2, and C_3 are the complex amplitude of the complex signal and ϕ_1, ϕ_2, and ϕ_3 correspond to the phase of the measured signal. We can further normalize Eq. (2) to the complex amplitude of the first waveguide,

$$U^{(p,q)} = \begin{pmatrix} 1 \\ \frac{C_2}{C_1} e^{i\phi_{12}^{(p,q)}} \\ \frac{C_3}{C_1} e^{i\phi_{13}^{(p,q)}} \end{pmatrix} e^{i(p\omega_1 + q\omega_2)t}. \tag{3}$$

A phi-bit with characteristic frequency $F^{(p,q)}$ can also be characterized by the phase difference of the measured displacement between waveguides 1 and 2 ($\phi_{12}^{(p,q)}$) and between waveguides 1 and 3 ($\phi_{13}^{(p,q)}$) by setting waveguide 1's phase ($\phi_1^{(p,q)}$) as the reference phase as shown below,

$$\begin{aligned} \phi_{12}^{(p,q)} &= \phi_1^{(p,q)} - \phi_2^{(p,q)}, \\ \phi_{13}^{(p,q)} &= \phi_1^{(p,q)} - \phi_3^{(p,q)}. \end{aligned} \tag{4}$$

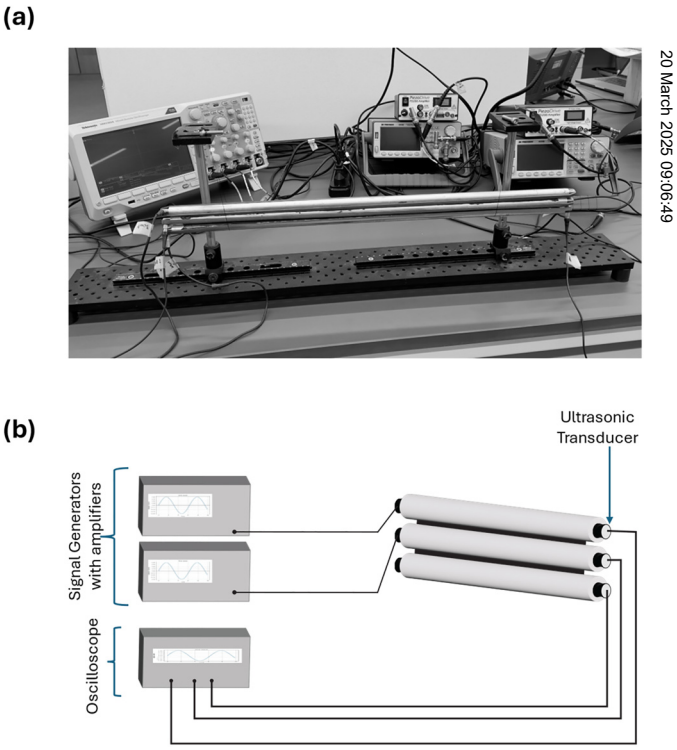

FIG. 1. (a) Picture of the experimental setup showing the array of elastically coupled waveguides, two signal generators and amplifiers on the right, and an oscilloscope on the left. (b) Schematic illustration of the experimental setup.

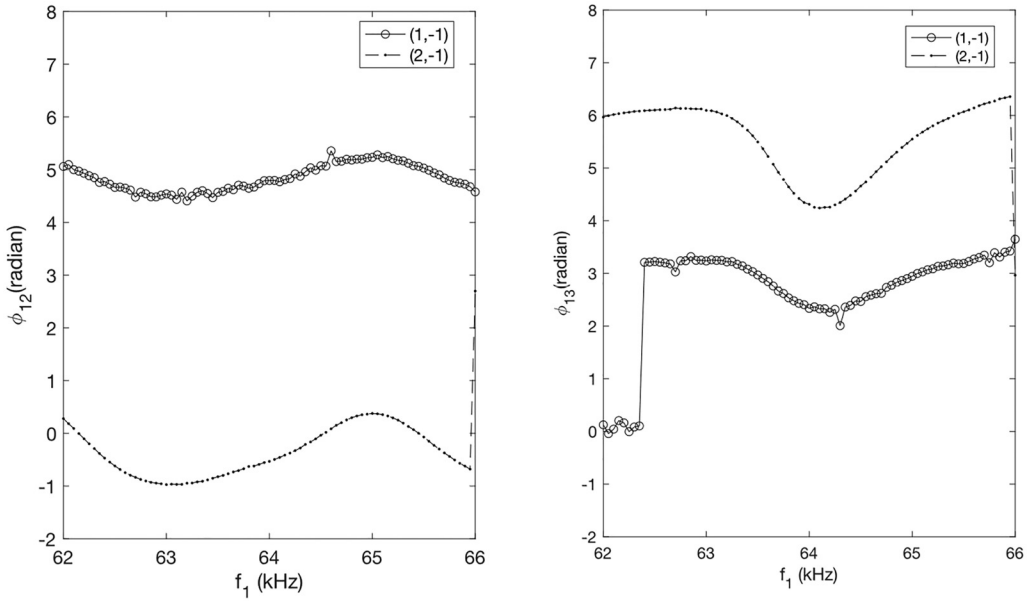

FIG. 2. Examples of observed phase differences $\phi_{12}^{(p,q)}(f_1)$ and $\phi_{13}^{(p,q)}(f_1)$ showing examples of continuous and π jump behavior. The values of p and q are given in the figure legend.

Dropping the direct reference to waveguide 1, we simplify Eq. (3) to

$$U^{(p,q)} = \begin{pmatrix} \hat{C}_2 e^{i\phi_{12}^{(p,q)}} \\ \hat{C}_3 e^{i\phi_{13}^{(p,q)}} \end{pmatrix} e^{i(p\omega_1 + q\omega_2)t}. \tag{5}$$

One can further expand the phi-bit displacement vector with an orthogonal basis $|0\rangle$ and $|1\rangle$, which represents the waveguides 2 and 3,

$$U^{(p,q)} = \hat{C}_2 e^{i\phi_{12}^{(p,q)}} |0\rangle + \hat{C}_3 e^{i\phi_{13}^{(p,q)}} |1\rangle. \tag{6}$$

The phase differences are controlled by varying the driving frequency f_1 by $\Delta\upsilon$ ranging from 0 to 4 kHz while keeping f_2 fixed. We track the location of the phi-bit on the spectral domain and compute the phase differences at each new phi-bit characteristic frequency.

The phase differences exhibit two consistent behaviors as a function of the driving frequency f_1. These behaviors are shown in Fig. 2. The phase differences are the superposition of monotonous continuous variations and sudden π jumps occurring at certain frequencies. The continuous behavior is consistent across all phi-bits. In particular, if we monitor the phase differences between rods 1 and 2 and rods 1 and 3 at the primary frequencies, namely, $\phi_{12}(f_i)$ and $\phi_{13}(f_i)$ where $i = 1,\ 2$, the monotonous contribution to $\phi_{12}^{(p,q)}$ and $\phi_{13}^{(p,q)}$ are given by $p\phi_{12}(f_1) + q\phi_{12}(f_2)$ and $p\phi_{13}(f_1) + q\phi_{13}(f_2)$. Both behaviors are consistently observed in all the phi-bits and used to achieve permutation operations described herein. We further note that these behaviors are general for phi-bits when driven at any frequencies f_1 and f_2, that is, the π jumps can be observed at different frequency f_1 for each phi-bits.

Each logical phi-bit characterized by its frequency possesses two independent degrees of freedom in the form of the measured phase differences (ϕ_{12} and ϕ_{13}). We make the choice of a Bloch's sphere representation of the state of a logical phi-bit (p, q) as follows:

$$V^{(p,q)} = \begin{pmatrix} \sin(\beta^{(p,q)}) \\ e^{i\gamma^{(p,q)}} \cos(\beta^{(p,q)}) \end{pmatrix}. \tag{7}$$

In Eq. (7), the Bloch sphere representation includes two degrees of freedom β and γ. These two degrees of freedom are taken as functions of ϕ_{12} and/or ϕ_{13}. One has great latitude in choosing these functions for each phi-bit in order to span its Bloch sphere; specifically, the phase difference for each phi-bit is determined by the multiplication factors p and q of the primary frequencies f_1 and f_2, as described in the continuous behavior of the system. This results in distinct phase differences for each phi-bit at different frequencies, providing a mechanism for independent control over the Bloch sphere parameters. Additionally, π jumps are observed in some of the phi-bits phase differences, providing additional control on parametrically spanning the Bloch.

In this work, we are able to realize all possible permutation gates acting on two phi-bits by taking $\gamma = 0$. In this case, the logical phi-bit has only real components, which is equivalent to a rebit.[20–22] β is also chosen as a function of ϕ_{12} or ϕ_{13}. With this,

the representation of Eq. (7) is simplified to

$$V^{(p,q)} = \begin{pmatrix} \sin(\beta^{(p,q)}) \\ \cos(\beta^{(p,q)}) \end{pmatrix}. \tag{8}$$

Consequently, the state vector of a two phi-bits system can be represented as the tensor product of each phi-bits state vector,

$$V = \begin{pmatrix} \sin(\beta^{(1)})\sin(\beta^{(2)}) \\ \sin(\beta^{(1)})\cos(\beta^{(2)}) \\ \cos(\beta^{(1)})\sin(\beta^{(2)}) \\ \cos(\beta^{(1)})\cos(\beta^{(2)}) \end{pmatrix}, \tag{9}$$

where $\beta^{(1)}$ and $\beta^{(2)}$ are associated to phi-bit 1 and phi-bit 2, respectively.

Even though the superposition of states given by Eq. (9) is a tensor product state, the nonlinear correlation between phi-bits 1 and 2 makes those components dependent on each other and, therefore, changes in the state of phi-bit 1 are related to changes in the state of phi-bit 2. This tensor product is limited to a subregion of the two phi-bit Hilbert space.

III. PERMUTATION GATES ACTING ON TWO PHI-BITS

To implement permutations with two phi-bits, we introduce a new representation of the two phi-bit state vectors,

$$\hat{V} = \begin{pmatrix} \sin(\beta^{(1)})\sin(\beta^{(2)}) \\ \frac{1}{2}[(1 - H(\cos(\beta^{(2)}))) * \sin(\beta^{(1)}) + (1 + H(\cos(\beta^{(2)}))) * \cos(\beta^{(1)})] * \cos(\beta^{(2)}) \\ \frac{1}{2}[(1 + H(\cos(\beta^{(2)}))) * \sin(\beta^{(1)}) + (1 - H(\cos(\beta^{(2)}))) * \cos(\beta^{(1)})] * \cos(\beta^{(2)}) \\ \cos(\beta^{(1)})\sin(\beta^{(2)}) \end{pmatrix}. \tag{10}$$

In Eq. (10), H(x) stands for the Heaviside function such that $\mathrm{H}(x) = \frac{x}{|x|} = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \end{cases}$. The representation of a two phi-bit state given by Eq. (10) is not in general a tensor product state and spans a wider region of the two phi-bit Hilbert space. The components of this vector are correlated with each other through the nonlinearity in the system.

For the sake of simplicity, we use a more compact notation such that $\sin(\beta^{(1)})$, $\cos(\beta^{(1)})$, $\sin(\beta^{(2)})$, and $\cos(\beta^{(2)})$ are written as S_1, C_1, S_2, and C_2, respectively. With this, Eq. (10) becomes

$$\hat{V} = \begin{pmatrix} S_1 S_2 \\ \frac{1}{2}[(1 - H(C_2)) * S_1 + (1 + H(C_2)) * C_1] * C_2 \\ \frac{1}{2}[(1 + H(C_2)) * S_1 + (1 - H(C_2)) * C_1] * C_2 \\ C_1 S_2 \end{pmatrix}. \tag{11}$$

Considering an initial vector $\hat{V} = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix}$, we are seeking changes in the driving conditions leading to changes in $\beta^{(1)}$ and $\beta^{(2)}$ such that the state vector transforms according to

$$V^* = P \times \hat{V}, \tag{12}$$

where P is a 4×4 permutation unitary matrix.

A. Realizing SWAP permutation gate

Here, we realize the permutation matrix, $P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, which permutes the second and third components of the two phi-bits state vector. This is the SWAP permutation gate. Let us consider a two phi-bit system initialized at arbitrary values of $\beta^{(1)}$ and $\beta^{(2)}$. Depending on the initial value of $\beta^{(2)}$ (i.e., the sign of the C_2), Eq. (11) gives two possible initial state vectors. We choose $C_2 > 0$ such that the initial state vector is taking the form

$$\hat{V} = \begin{pmatrix} S_1 S_2 \\ C_1 C_2 \\ S_1 C_2 \\ C_1 S_2 \end{pmatrix}. \tag{13}$$

To swap components 2 and 3 of this state vector, we need to tune the frequency of the physical system such that the phase difference of phi-bit 1 (either ϕ_{12} or ϕ_{13} defining $\beta^{(1)}$) remains constant and the phase difference of phi-bit 2 (defining $\beta^{(2)}$) changes by $\pm\pi$ such as

$$\begin{aligned} \beta^{(1)*} &= \beta^{(1)}, \\ \beta^{(2)*} &= \beta^{(2)} \pm \pi. \end{aligned} \tag{14}$$

In Eq. (14), the * labels the phase difference after the change in frequency. The change by π of $\beta^{(2)}$ changes the sign of C_2 to a negative value such that Eq. (11) gives a new state vector through

the Heaviside function. As a result, the state vector transforms according to

$$\hat{V} = \begin{pmatrix} S_1S_2 \\ C_1C_2 \\ S_1C_2 \\ C_1S_2 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -S_1S_2 \\ -S_1C_2 \\ -C_1C_2 \\ -C_1S_2 \end{pmatrix}. \tag{15}$$

The minus signs come from the trigonometric relations $\cos(\beta^{(2)} \pm \pi) = -\cos(\beta^{(2)})$ and $\sin(\beta^{(2)} \pm \pi) = -\sin(\beta^{(2)})$.

To a general phase of π associated with the – sign in V^* that we can disregard, the transformation can be expressed in terms of the desired SWAP gate as follows:

$$V^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ C_1C_2 \\ S_1C_2 \\ C_1S_2 \end{pmatrix} = \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix}. \tag{16}$$

To execute the SWAP gate experimentally, we consider $\beta^{(i)}$, $i = 1, 2$ to be the simplest function of $\phi_{12}^{(p,q)}$ by taking $\beta^{(i)} = \phi_{12}^{(p(i),q(i))}$. We choose phi-bit 1 to be the nonlinear mode with $(p = 3,\ q = -2)$ and phi-bit 2 to be the mode with $(p = 3,\ q = -1)$. The phase difference of phi-bit 1 $(\phi_{12}^{(3,-2)})$ and phi-bit 2 $(\phi_{12}^{(3,-1)})$ is illustrated in Fig. 3. We initialized the phi-bits at frequency $f_1 = 64.05$ kHz and tune the driving frequency to 64.15 kHz. At the initial frequency, the initial ϕ_{12} of phi-bit 1 is 2.04 rad and the initial ϕ_{12} of phi-bit 2 is 5.94 rad. After tuning, the phase difference for phi-bit 1 remains nearly constant and the phase difference for phi-bit 2 decreases by π through the π jump. The change in state vectors for this permutation is shown below where considering the absolute value of the state vector; we observed the second and third components of the state vector are swapped,

$$\hat{V} = \begin{pmatrix} -0.30 \\ -0.43 \\ 0.84 \\ 0.15 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} 0.20 \\ -0.82 \\ 0.52 \\ -0.12 \end{pmatrix}. \tag{17}$$

Disregarding the change in signs, the physical operation swaps the second component with the third component while keeping the first and the fourth components nearly the same. We note that the numerical values of the components of the state vector may change by no more than 0.1 due to the fact that the jump in phase is not exactly π and that $\beta^{(1)}$ is not exactly constant over the range of frequency spanned by the physical operation.

In Eq. (13), the value of the component of the initial vector is determined by the constant functions $\beta^{(1)} = \phi_{12}^{(3,-2)}$ and $\beta^{(2)} = \phi_{12}^{(3,-1)}$ and the specific value of the phases at the frequency $f_1 = 64.05$ kHz. However, one can span the complete unit circle of phi-bit 1 by choosing the function,

$$\beta^{(1)} = \phi_{12}^{(3,-2)} + \alpha^{(1)}, \tag{18}$$

where $\alpha^{(1)} \in [-\pi, \pi]$. Similarly, one can span the complete unit circle of phi-bit 2 by choosing the function

$$\beta^{(2)} = \phi_{12}^{(3,-1)} + \alpha^{(2)}, \tag{19}$$

where $\alpha^{(2)} \in [-\pi, \pi]$. By choosing the value of $\alpha^{(1)}$ and $\alpha^{(2)}$, one can generate any tensor product state of the two phi-bits as input to the swap gate. The functions $\beta^{(i)} = \phi_{12}^{(p(i),q(i))} + \alpha^{(i)}$, $i = 1,2$ enable us to select any input vector $\hat{V}$ without losing the ability of swapping the components by operating the following π jump, $\beta^{(2)} \pm \pi$ since the π jump only affects the $\phi_{12}^{(p,q)}$.

B. Realizing C-NOT permutation gate

Here, we realize the permutation matrix, $P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$, which permutes the third and fourth components of the two phi-bits state vector. This is the C-NOT gate. In this case, we use the same initial state vector as in Subsection III A [Eq. (13)] but must restrict the initial phase difference such that $\beta^{(1)} \in [0,\ \pi]$ and $\beta^{(2)} \in \left[\frac{-\pi}{2},\ \frac{\pi}{2}\right]$.

By tuning the driving frequency, such that the phase difference of the first phi-bit exchange with that of the second phi-bit,

$$\begin{aligned} \beta^{(1)*} &= \beta^{(2)}, \\ \beta^{(2)*} &= \beta^{(1)}, \end{aligned} \tag{20}$$

the transformed state vector becomes by permutation of indices: $V^* = \begin{pmatrix} S_2S_1 \\ C_2C_1 \\ S_2C_1 \\ C_2S_1 \end{pmatrix}$. This transformation is equivalent to the unitary transformation of the state vector with a C-NOT gate by commutativity of the multiplication,

$$V^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ C_1C_2 \\ S_1C_2 \\ C_1S_2 \end{pmatrix} = \begin{pmatrix} S_1S_2 \\ C_1C_2 \\ C_1S_2 \\ S_1C_2 \end{pmatrix}. \tag{21}$$

We set $\beta^{(i)}$, $i = 1,\ 2$ as a simple function of $\phi_{13}^{(p,q)}$ only by taking $\beta^{(i)} = \phi_{13}^{(p(i),q(i))}$. We implement the C-NOT permutation using experimentally measured data for phi-bit 1 $(\phi_{13}^{(1,-2)})$ and phi-bit 2 $(\phi_{13}^{(-1,2)})$. These phases are shown in Fig. 4. The experimental phase difference for phi-bit 1 lies between 0 and π. The experimental phase difference for phi-bit 2 was translated by an inconsequential constant ε to constrain it to be between $\frac{-\pi}{2}$ and $\frac{\pi}{2}$. This translation is of the form $\beta^{(2)} = \phi_{13}^{(-1,2)} + \varepsilon$. The C-NOT permutation is achieved by initializing the phi-bits at frequency $f_1 = 63.15$ kHz and tuning it to 63.70 kHz. The initial ϕ_{13} for phi-bit 1 is 0.76 rad, and the initial ϕ_{13} for phi-bit 2 is 0.31 rad. After frequency tuning, the phase differences of phi-bit 1 and phi-bit 2 are exchanged so that the final phase difference for phi-bit 1 is the initial phase difference for phi-bit 2 and the final phase difference for phi-bit 2 is the initial phase difference

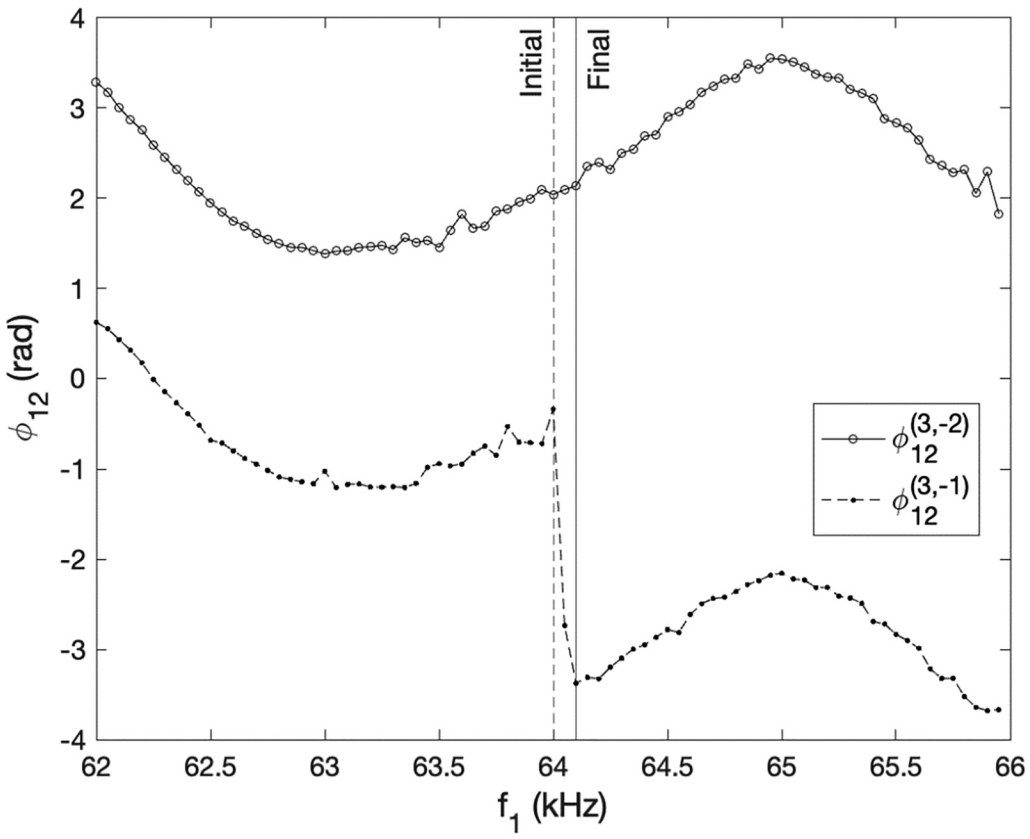

FIG. 3. Experimentally measured ϕ_{12} of phi-bit 1 $\phi_{12}^{(3,-2)}$ and phi-bit 2 $\phi_{12}^{(3,-1)}$ for implementing the SWAP gate by tuning the driving frequency from an initial marked by the vertical dashed line to a final frequency indicated by a solid vertical line.

for phi-bit 1. The initial and final state vectors are shown below where we observed the third and fourth components are swapped,

$$\hat{V} = \begin{pmatrix} 0.21 \\ 0.70 \\ 0.65 \\ 0.22 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} 0.19 \\ 0.69 \\ 0.19 \\ 0.68 \end{pmatrix}. \tag{22}$$

After permutation, the small changes in the numerical values of the components (less than 5%) arise from the small amount of noise in the experimentally measured phases.

In Eq. (22), the value of the component of the initial vector is determined by the constant functions $\beta^{(1)} = \phi_{13}^{(1,-2)} = 0.76$ and $\beta^{(2)} = \phi_{13}^{(-1,2)} + \varepsilon = 0.31$ at the frequency $f_1 = 63.15$ kHz. It is important to note that in the operation described above we have considered two stages. The first stage is the initialization of the state of the phi-bits, that is, defining the input vector $\hat{V}$. The second stage corresponds to the C-NOT operation itself, which transforms the input vector into an output vector, V^*. In Eq. (18), we have illustrated these two stages in a specific case; however, there exists an infinite number of possible input vectors, which can be operated upon. These inputs, which conserve the symmetry of the crossing occurring at the frequency $f_1 = 63.4$ kHz, form a subspace of the Hilbert space of all possible states. One can generate these inputs by using $\beta^{(1)}(f_1) = \delta(\phi_{13}^{(1,-2)}(f_1) - \phi_{13}^{(1,-2)}(63.4\,\text{kHz}))$ and $\beta^{(2)}(f_1) = \delta(\phi_{13}^{(-1,2)}(f_1) + \varepsilon - \phi_{13}^{(1,-2)}(63.4\,\text{kHz}))$ for $f_1 \in [63.15,\ 63.70\,\text{kHz}]$. Here, we have translated the crossing point such that it occurs at the origin of phases. The rescaling factor δ is used to span the complete range of angles. A second infinite set of possible input states can be generated by using the transformations $\beta^{(1)}(f_1) = \delta(\phi_{13}^{(1,-2)}(f_1) - \phi_{13}^{(1,-2)}(63.4\text{kHz})) + \alpha$ and $\beta^{(2)} = \delta(\phi_{13}^{(-1,2)}(f_1) + \varepsilon - \phi_{13}^{(1,-2)}(63.4\text{kHz})) + \alpha$, where α translates the crossing after rescaling. The C-NOT operation can subsequently operate on these infinite sets of inputs. To operate on states within the complete Hilbert space, one needs to implement another initialization step.

First one chooses one of the input states that conserves the symmetry of the crossing within the infinite set of such states, for example, let us consider one of the possible symmetry-conserving input state, $\beta^{(1)} = 0.76$ and $\beta^{(2)} = 0.31$. This symmetry-conserving state is redefined as a target state ($\beta_t^{(1)} = 0.76$ and $\beta_t^{(2)} = 0.31$).

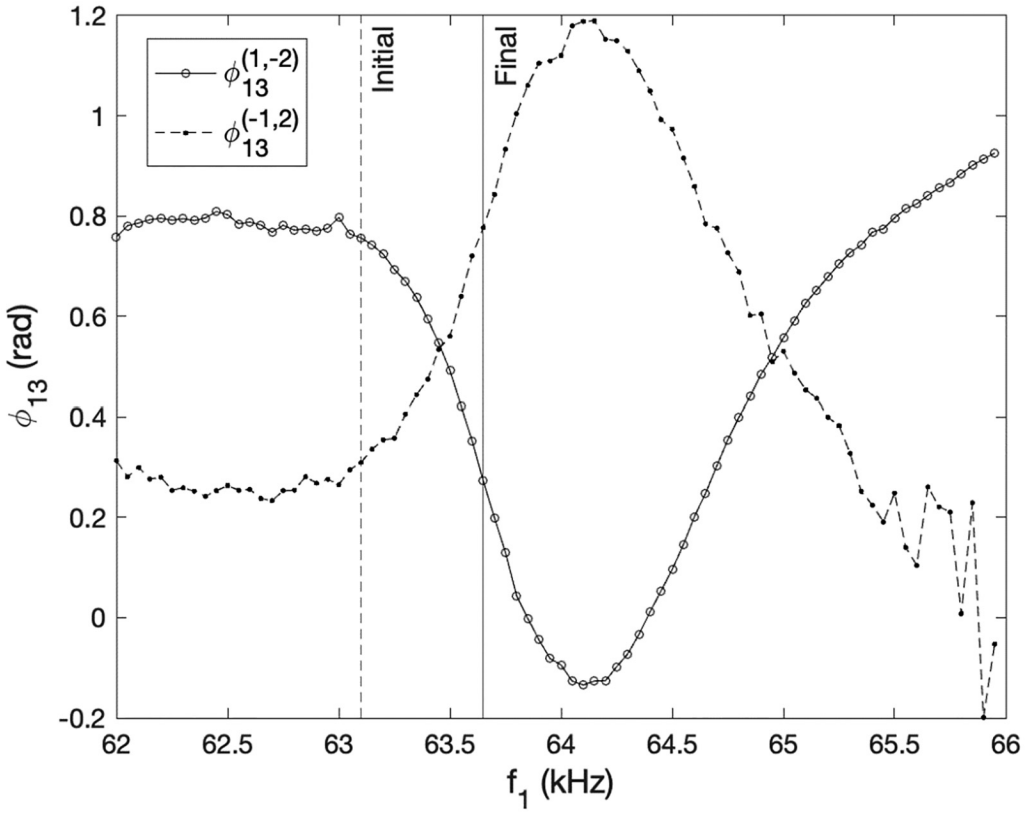

FIG. 4. Experimentally measured ϕ_{13} of phi-bit 1 $\phi_{13}^{(1,-2)}$ and phi-bit 2 $\phi_{13}^{(-1,2)}$ for implementing the C-NOT gate as a function of driving frequency. Vertical lines (dashed and solid) as in Fig. 3.

One now needs to map a desired input to the target input state. The mapping from a desired state ($\beta_d^{(1)}$ and $\beta_d^{(2)}$, which can be used to span the complete unit circle) to the target state can be done with the functions $\beta_t^{(1)} = \beta_d^{(1)} + \alpha'^{(1)}$, where $\alpha'^{(1)} = \phi_{13}^{(1,-2)} - \beta_d^{(1)}$, and $\beta_t^{(2)} = \beta_d^{(2)} + \alpha'^{(2)}$, where $\alpha'^{(2)} = \phi_{13}^{(-1,2)} - \beta_d^{(2)}$. However, this mapping is not invertible. Since one has great latitude in selecting functions that can be used to realize the mapping, another possibility, without using translations is to employ a nearly constant function to create this mapping. For instance, one can use the mapping $\beta_t^{(1)} \cos\mu\beta_d^{(1)}$ and $\beta_t^{(2)} \cos\mu\beta_d^{(2)}$, where μ is a very small quantity such that the cosine function is nearly constant over the complete unit circle. With such a function, the level of accuracy is determined by the parameter μ. The advantage of this type of mapping is its invertibility. We emphasize that the stage of the initialization of the input state is separate from the stage of the C-NOT operation. It is also important to note that the mapping between the desired state and the target state is taking place in the space of the $\beta_t^{(i)}$ and $\beta_d^{(i)}$, which scales linearly with the number of phi-bits. Consequently, initialization does not cost significant overhead. After initialization, the C-NOT gate can be applied to (a) the input vector associated with symmetry-conserving states or (b) any other input vector obtained through mapping to a symmetry-conserving state, by tuning the frequency. We emphasize that the C-NOT gate operates in the tensor product space of the phi-bits. It is this stage which gives an advantage to operating with a phi-bit-based approach. This two stage technique is general and employed throughout the manuscript for exploring permutations.

C. Permuting components 1 and 2

Let us use the initial state vector of Subsection III B with the same restrictions ($\beta^{(1)} \in [0, \pi]$ and $\beta^{(2)} \in [\frac{-\pi}{2}, \frac{\pi}{2}]$).

To achieve permutation of components 1 and 2 of the state vector, we must tune the frequency of the physical system such that $\beta^{(1)}$ is exchanged with the $\beta^{(2)}$ and $\beta^{(1)}$ increases by $\frac{\pi}{2}$ and $\beta^{(2)}$ decreases by $\frac{\pi}{2}$,

$$\begin{aligned} \beta^{(1)*} &= \beta^{(2)} + \frac{\pi}{2}, \\ \beta^{(2)*} &= \beta^{(1)} - \frac{\pi}{2}. \end{aligned} \tag{23}$$

Using the exchange of indices and the trigonometric relations, $\cos\left(\theta+\frac{\pi}{2}\right)=-\sin(\theta)$, $\sin\left(\theta+\frac{\pi}{2}\right)=\cos(\theta)$, $\cos\left(\theta-\frac{\pi}{2}\right)=\sin(\theta)$, and $\sin\left(\theta-\frac{\pi}{2}\right)=-\cos(\theta)$, the initial state vector transforms according to

$$\hat{V}=\begin{pmatrix} S_1S_2 \\ C_1C_2 \\ S_1C_2 \\ C_1S_2 \end{pmatrix} \rightarrow V^*=\begin{pmatrix} -C_2C_1 \\ -S_2S_1 \\ C_2S_1 \\ S_2C_1 \end{pmatrix}. \tag{24}$$

There is a π phase difference between some of the components of the new state vector due to the sign change. This operation is described below,

$$V^*=\begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} S_1S_2 \\ C_1C_2 \\ S_1C_2 \\ C_1S_2 \end{pmatrix}=\begin{pmatrix} -C_1C_2 \\ -S_1S_2 \\ S_1C_2 \\ C_1S_2 \end{pmatrix}. \tag{25}$$

The transformation matrix in Eq. (25) is the product of a phase gate and the desired permutation gate, such that

$$\begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}=\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Disregarding the change in sign, we have achieved the permutation of the first and second components.

This permutation is achieved by expressing $\beta^{(i)}$, $i=1,\ 2$ as a simple function of $\phi_{13}^{(p,q)}$ such that $\beta^{(i)}=\phi_{13}^{(p(i),q(i))}$ and using experimentally measured data of phi-bit 1 ($\phi_{13}^{(1,-2)}$) and phi-bit 2 ($\phi_{13}^{(-1,2)}$), as shown in Fig. 5. For this permutation, the experimental phase difference of phi-bit 1 is translated by a constant phase such that it lies between 0 and π. The experimental phase difference of phi-bit 2 is translated by another constant phase ε such that the phase difference falls between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. This translation is of the form $\beta^{(2)}=\phi_{13}^{(-1,2)}+\varepsilon$. The phi-bits are initialized at frequency $f_1=63.45\,\text{kHz}$, and the frequency of the system is tuned from its initial value to 63.95 kHz. At the initial frequency, the initial ϕ_{13} for phi-bit 1 is 1.12 rad and the initial ϕ_{13} of phi-bit 2 is -1.08 rad. After tuning, the phase difference for phi-bit 1 becomes the initial phase difference of phi-bit 2 plus $\frac{\pi}{2}$ and phase difference of phi-bit 2 becomes the initial phase difference of phi-bit 1 minus $\frac{\pi}{2}$. This gives the change in state vectors shown below,

$$\hat{V}=\begin{pmatrix} -0.80 \\ 0.20 \\ 0.41 \\ -0.39 \end{pmatrix} \rightarrow V^*=\begin{pmatrix} -0.21 \\ 0.80 \\ 0.41 \\ -0.40 \end{pmatrix}. \tag{26}$$

Making abstraction of the change in sign, V^* is obtained by simply permuting components 1 and 2 of V.

The small variations in numerical values of the state vector components result from small noise in the experimental data.

In Eq. (26), the value of the component of the initial vector, fixed input state, is determined by the constant functions $\beta^{(1)}=\phi_{13}^{(1,-2)}=1.12$ and $\beta^{(2)}=\phi_{13}^{(-1,2)}+\varepsilon=-1.08$ at the frequency $f_1=63.45\,\text{kHz}$. An approach similar to that of Sec. III B can be used to create any desired input state by mapping it to a fixed input state (or any symmetry-conserving state). Here, we require that $\beta^{(1)}$ and $\beta^{(2)}$ maintain a $\frac{\pi}{2}$ change and a consistent separation as we translate the state. One can initialize the states by using $\beta^{(1)}(f_1)=\phi_{13}^{(1,-2)}(f_1)+\alpha$ and $\beta^{(2)}(f_1)=\phi_{13}^{(-1,2)}(f_1)+\varepsilon+\alpha$ for $f_1\in[62.00,\ 63.50\,\text{kHz}]$ to generate any input for the permutation process. The translation factor α is used to select a subspace of the Hilbert space of all the infinitely possible states and maintain spacing between the $\beta^{(i)}$, where $i=1,\ 2$. This separation-conserving state can then be mapped to any desired input state using the same invertible mapping approach described in Sec. III B, with $\beta_t^{(1)}\cos\mu\beta_d^{(1)}$ and $\beta_t^{(2)}\cos\mu\beta_d^{(2)}$, where μ is a very small quantity such that the cosine function is nearly constant over the complete unit circle. After the initialization stage which takes place in the linearly scaling space of phases, one can apply a change in tuning frequency in a separate stage to permute components 1 and 2 of any input state whether it is a separation-conserving state vector or an input vector obtained by mapping to a separation-conserving state. Again, we emphasize that this permutation gate operates in the tensor product space of the phi-bits state.

D. Permuting components 1 and 3

We consider the initial state vector of a two phi-bits system with the following restrictions on $\beta^{(1)}$ and $\beta^{(2)}$: $\beta^{(1)}\in[\pi,2\pi]$ and $\beta^{(2)}\in\left[\frac{\pi}{2},\frac{3\pi}{2}\right]$. With this $C_2<0$ and Eq. (11) gives the initial state vector below,

$$\hat{V}=\begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix}. \tag{27}$$

To permute components 1 and 3 of the state vector, we must tune the frequency of the physical system such that $\beta^{(1)}$ is exchanged with the $\beta^{(2)}$ and $\beta^{(1)}$ increases by $\frac{\pi}{2}$ and $\beta^{(2)}$ decreases by $\frac{\pi}{2}$ as was also done in Subsection III C:

$$\begin{aligned} \beta^{(1)*}&=\beta^{(2)}+\frac{\pi}{2}, \\ \beta^{(2)*}&=\beta^{(1)}-\frac{\pi}{2}. \end{aligned} \tag{28}$$

By permuting the indices and using the same trigonometric relations as in Subsection III C, the state vector transforms

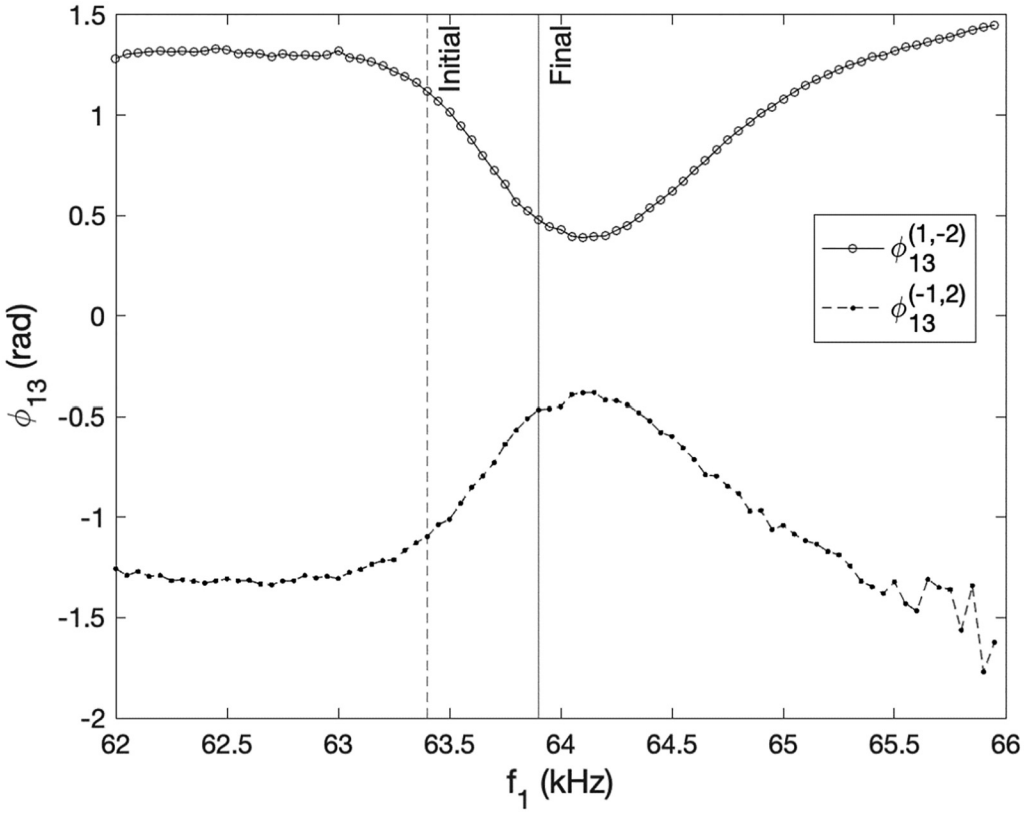

FIG. 5. Experimentally measured ϕ_{13} of phi-bit 1 ($\phi_{13}^{(1,-2)}$) and phi-bit 2 ($\phi_{13}^{(-1,2)}$) for implementing the permutation of components 1 and 2. Vertical lines (dashed and solid) as in Fig. 3.

according to

$$\hat{V} = \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -C_2C_1 \\ C_2S_1 \\ -S_2S_1 \\ S_2C_1 \end{pmatrix}. \tag{29}$$

This transformation permutes the first and third components and is equivalent to a transformation achieved using the matrix given below,

$$V^* = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} = \begin{pmatrix} -C_1C_2 \\ S_1C_2 \\ -S_1S_2 \\ C_1S_2 \end{pmatrix}. \tag{30}$$

The transformation matrix in Eq. (30) is the product of a phase gate and the desired permutation gate, such that

$$\begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

By disregarding the phase change in the transformed state vector V^*, we have successfully permuted the first and third components of the state vector.

This permutation is achieved with the same two phi-bits as in Subsection III C. However, the experimental phase difference of phi-bit 1 was adjusted by a constant to fall between π and 2π as shown in Fig. 6. The experimental phase difference of phi-bit 2 was translated by another constant ε to lie between $\frac{\pi}{2}$ and $\frac{3\pi}{2}$. This translation is of the form $\beta^{(2)} = \phi_{13}^{(-1,2)} + \varepsilon$. The phi-bits are now initialized at frequency $f_1 = 63.45$ kHz. At the initial frequency of 63.45 kHz, the initial ϕ_{13} of phi-bit 1 is 4.61 rad and the initial ϕ_{13} of phi-bit 2 is 2.39 rad. This permutation is achieved by tuning the frequency to 63.95 kHz. Upon tuning, the phase difference of phi-bit 1 becomes the initial phase difference of phi-bit 2 plus $\frac{\pi}{2}$ and the phase difference of phi-bit 2 becomes the initial phase difference of phi-bit 1 minus $\frac{\pi}{2}$. The resulting change in state vectors

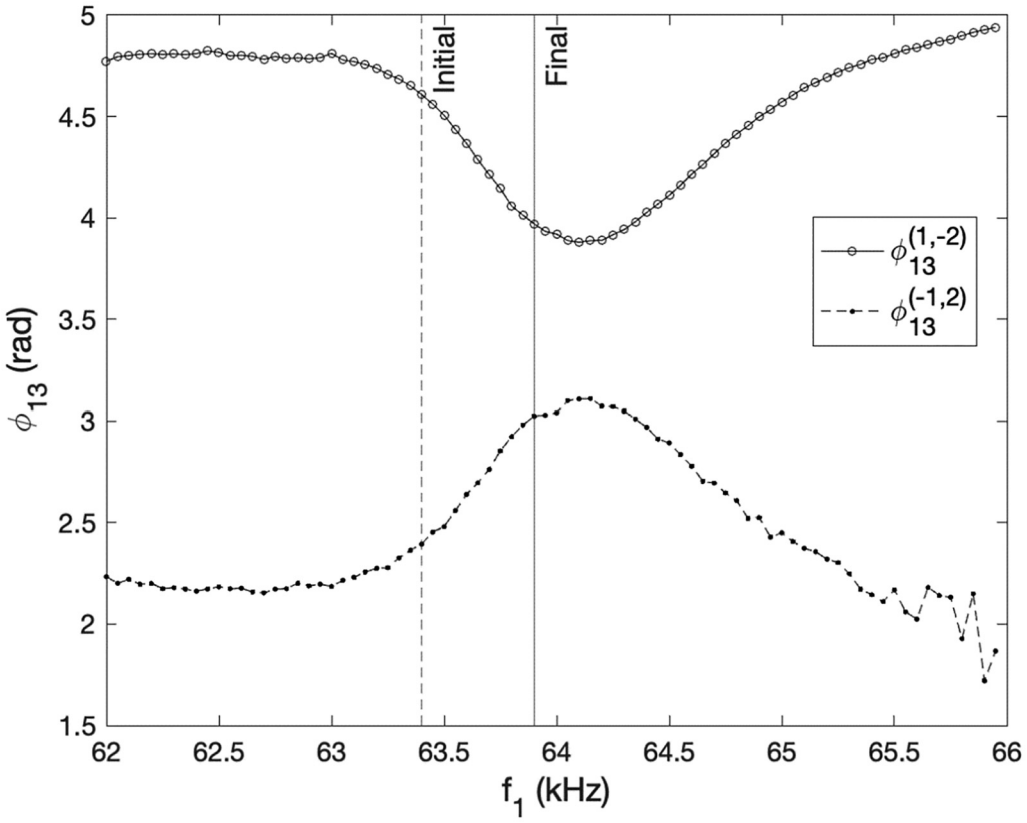

FIG. 6. Experimentally measured ϕ_{13} of phi-bit 1 ($\phi_{13}^{(1,-2)}$) and phi-bit 2 ($\phi_{13}^{(-1,2)}$) for implementing the permutation of components 1 and 3. Vertical lines (dashed and solid) as in Fig. 3.

is illustrated below,

$$\hat{V} = \begin{pmatrix} -0.68 \\ 0.73 \\ 0.08 \\ -0.07 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -0.09 \\ 0.73 \\ 0.67 \\ -0.08 \end{pmatrix}. \tag{31}$$

Independently of some changes in sign, this transformation permutes components 1 and 3 with minimal change in the numerical value of the components themselves.

In Eq. (31), the value of the component of the initial vector, fixed input state, is determined by the constant functions $\beta^{(1)} = \phi_{13}^{(1,-2)} = 4.61$ and $\beta^{(2)} = \phi_{13}^{(-1,2)} + \varepsilon = 2.39$ at frequency $f_1 = 63.45$ kHz. However, one can span the complete unit circle of phi-bit 1 and phi-bit 2 by using the same approach described in Sec. III B, which maps the desired input to the fixed input state.

E. Permuting components 1 and 4

For the permutation of components 1 and 4, let us consider a two phi-bit system initialized at arbitrary values of $\beta^{(1)}$ and $\beta^{(2)}$. Like Subsection III A, Eq. (11) gives two possible initial state vectors depending on the initial value of $\beta^{(2)}$ (i.e., the sign of the C_2). Here, we choose $C_2 < 0$ such that the initial state vector takes the form

$$\hat{V} = \begin{pmatrix} S_1 S_2 \\ S_1 C_2 \\ C_1 C_2 \\ C_1 S_2 \end{pmatrix}. \tag{32}$$

Tuning the frequency to manipulate the phase difference such that $\beta^{(1)}$ increases by $\frac{\pi}{2}$ and $\beta^{(2)}$ by π,

$$\begin{aligned} \beta^{(1)*} &= \beta^{(1)} + \frac{\pi}{2}, \\ \beta^{(2)*} &= \beta^{(2)} + \pi. \end{aligned} \tag{33}$$

Adding π to $\beta^{(2)}$ makes $C_2 > 0$, by virtue of the trigonometric relation $\cos(\theta + \pi) = -\cos(\theta)$, this affects the value of the Heaviside function in Eq. (11) such that the vector V becomes $\begin{pmatrix} -S_1 S_2 \\ C_1 C_2 \\ S_1 C_2 \\ -C_1 S_2 \end{pmatrix}$. The minus signs arise from the trigonometric relation

$\sin(\theta + \pi) = -\sin(\theta)$. This first step effectively swaps components 2 and 3. Using the trigonometric relations employed in Sec. III C to address the change in $\beta^{(1)}$ by $\frac{\pi}{2}$ transforms the state vector as

$$\hat{V} = \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -C_1S_2 \\ -S_1C_2 \\ C_1C_2 \\ S_1S_2 \end{pmatrix}. \tag{34}$$

In matrix form, the whole operation takes the form

$$V^* = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} = \begin{pmatrix} -C_1S_2 \\ -S_1C_2 \\ C_1C_2 \\ S_1S_2 \end{pmatrix}. \tag{35}$$

The unitary transformation in Eq. (35) is effectively the product of a phase gate with the desired permutation gate as follows:

$$\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

To within phase changes of the components of the state vector, we have realized the gate that permutes the first and the fourth components of the state vector.

This permutation is achieved by expressing $\beta^{(i)}$, $i = 1,\ 2$ as a simple function of $\phi_{12}^{(p,q)}$ and by taking $\beta^{(i)} = \phi_{12}^{(p(i),q(i))}$ using experimentally measured data of phi-bit 1 ($\phi_{12}^{(3,-2)}$) and phi-bit 2 ($\phi_{12}^{(-6,-3)}$), as shown in Fig. 7. Here, there is no restriction on the range of values of the phase differences, and we used the raw experimental data without adjustment. The permutation is achieved by initializing the phi-bits at frequency $f_1 = 63.35$ kHz and tuning it to 64.65 kHz. At the initial frequency, the initial ϕ_{12} for phi-bit 1 is 1.43 rad and the initial ϕ_{12} for phi-bit 2 is 0.34 rad. After tuning, the phase difference for phi-bit 1 increases by $\frac{\pi}{2}$ and the phase difference for phi-bit 2 increases by π. Increasing the phase difference of phi-bit 1 by $\frac{\pi}{2}$ reverses the order of the state vector, by permuting simultaneously components 2 and 3 and components 1 and 4. Increasing the phase difference for phi-bit 2 by π permutes again components 2 and 3, leaving just the permutation of components 1 and 4. The change in state vectors for this permutation is shown below,

$$\hat{V} = \begin{pmatrix} 0.33 \\ 0.13 \\ 0.93 \\ 0.05 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -0.03 \\ -0.10 \\ 0.94 \\ 0.32 \end{pmatrix}. \tag{36}$$

Considering the absolute value of the state vector components, we observed the permutation of the first and fourth components of that vector. Again, small variations in numerical values result from the experimental noise.

In Eq. (36), the value of the component of the initial vector, fixed input state, is determined by the constant functions $\beta^{(1)} = \phi_{12}^{(1,-2)} = 1.43$ and $\beta^{(2)} = \phi_{12}^{(-1,2)} = 0.34$ at frequency $f_1 = 63.35$ kHz. However, one can span the complete unit circle of phi-bit 1 and phi-bit 2 by using the same approach described in Sec. III B, which maps the desired input to the fixed input state.

F. Permuting components 2 and 4

We consider the initial state vector of a two phi-bits system with the same restrictions on $\beta^{(1)}$ and $\beta^{(2)}$ as in Subsection III D: $\beta^{(1)} \in [\pi, 2\pi]$ and $\beta^{(2)} \in [\frac{\pi}{2}, \frac{3\pi}{2}]$. Equation (11) gives the same initial state vector as in Eq. (27). The frequency of the driver must be tuned such that one exchanges $\beta^{(1)}$ and $\beta^{(2)}$,

$$\begin{aligned} \beta^{(1)*} &= \beta^{(2)}, \\ \beta^{(2)*} &= \beta^{(1)}. \end{aligned} \tag{37}$$

This tuning transforms the state vector as shown below,

$$\hat{V} = \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} S_2S_1 \\ S_2C_1 \\ C_2C_1 \\ C_2S_1 \end{pmatrix}. \tag{38}$$

This transformation permutes the second and fourth components of the state vector, which is equivalent to the unitary transformation using the matrix below,

$$V^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} = \begin{pmatrix} S_1S_2 \\ C_1S_2 \\ C_1C_2 \\ S_1C_2 \end{pmatrix}. \tag{39}$$

This permutation is achieved by expressing $\beta^{(i)}$, $i = 1,\ 2$ as a simple function of $\phi_{13}^{(p,q)}$ only by taking $\beta^{(i)} = \phi_{13}^{(p(i),q(i))}$ and using experimentally measured data of phi-bit 1 ($\phi_{12}^{(1,-2)}$) and phi-bit 2 ($\phi_{12}^{(-1,2)}$), as shown in Fig. 8. For this permutation, the experimental phase difference for phi-bit 1 was translated with a constant to constrain the value between π and 2π. The experimental phase difference for phi-bit 2 was adjusted by a constant ε such that the value falls between $\frac{\pi}{2}$ and $\frac{3\pi}{2}$. This translation is of the form $\beta^{(2)} = \phi_{13}^{(-1,2)} + \varepsilon$. The permutation is achieved by initializing the phi-bits at frequency $f_1 = 63.15$ kHz and tuning the frequency to 63.70 kHz. The initial ϕ_{13} for phi-bit 1 is 3.72 rad, and the initial ϕ_{13} for phi-bit 2 is 3.28 rad. After this tuning, the phase differences of phi-bit 1 and phi-bit 2 are swapped so that the final phase difference for phi-bit 1 is the initial phase difference for phi-bit 2 and the final phase difference for phi-bit 2 is the initial phase difference for phi-bit 1. The change in state vectors is shown below where we observed the second and fourth components of the state vector are

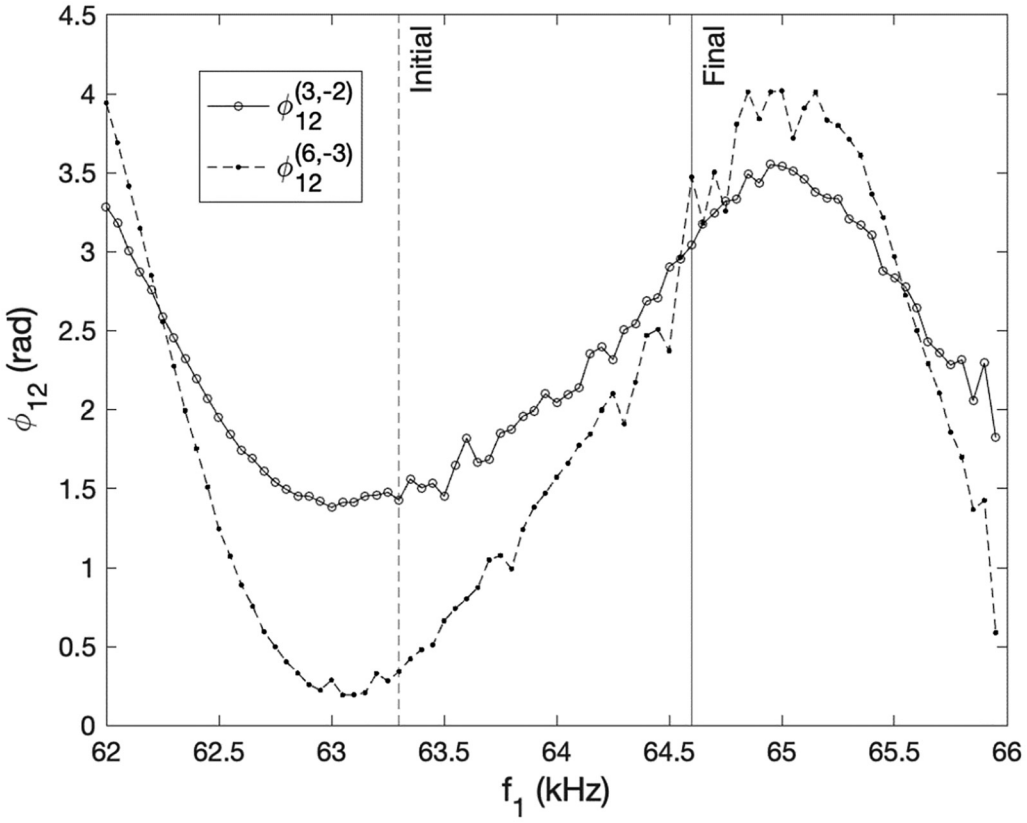

FIG. 7. Experimentally measured ϕ_{12} of phi-bit 1 ($\phi_{12}^{(3,-2)}$) and phi-bit 2 ($\phi_{12}^{(-6,-3)}$) for implementing the permutation of components 1 and 4. Vertical lines (dashed and solid) as in Fig. 3.

permuted,

$$\hat{V} = \begin{pmatrix} 0.07 \\ 0.54 \\ 0.83 \\ 0.11 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} 0.06 \\ 0.08 \\ 0.82 \\ 0.56 \end{pmatrix}. \tag{40}$$

Noise in the experimental data induces small variations in the numerical values of the initial and permuted state vectors.

In Eq. (40), the value of the component of the initial vector, fixed input state, is determined by the constant functions $\beta^{(1)} = \phi_{12}^{(1,-2)} = 3.72$ and $\beta^{(2)} = \phi_{12}^{(-1,2)} = 3.28$ at frequency $f_1 = 63.15$ kHz. However, one can apply this permutation to input states in the complete Hilbert space by using the same approach described in Sec. III B, which maps the desired input to the fixed input state.

G. Cyclic permutation or downward shift permutation

For a cyclic permutation where all the elements of a state vector are shifted downwards, we use the same initial state vector as in Subsection III D [Eq. (27)] but must restrict the initial phase difference such that $\beta^{(1)}$ and $\beta^{(2)}$ are $\in [\frac{\pi}{2}, \frac{3\pi}{2}]$. The driving frequency of the system must then be tuned such that the phase difference $\beta^{(1)}$ is exchanged with the $\beta^{(2)}$ and $\beta^{(2)}$ increases by $\frac{\pi}{2}$,

$$\begin{aligned} \beta^{(1)*} &= \beta^{(2)}, \\ \beta^{(2)*} &= \beta^{(1)} + \frac{\pi}{2}. \end{aligned} \tag{41}$$

This tuning transforms the state vector as shown below,

$$\hat{V} = \begin{pmatrix} S_1 S_2 \\ S_1 C_2 \\ C_1 C_2 \\ C_1 S_2 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} S_2 C_1 \\ -S_2 S_1 \\ -C_2 S_1 \\ C_2 C_1 \end{pmatrix}. \tag{42}$$

The minus signs arise from the trigonometric relation $\cos\left(\theta + \frac{\pi}{2}\right) = -\sin(\theta)$ and $\sin\left(\theta + \frac{\pi}{2}\right) = \cos(\theta)$. This transformation shifts the first three components down and brings the last components to the first index, which is equivalent to the unitary

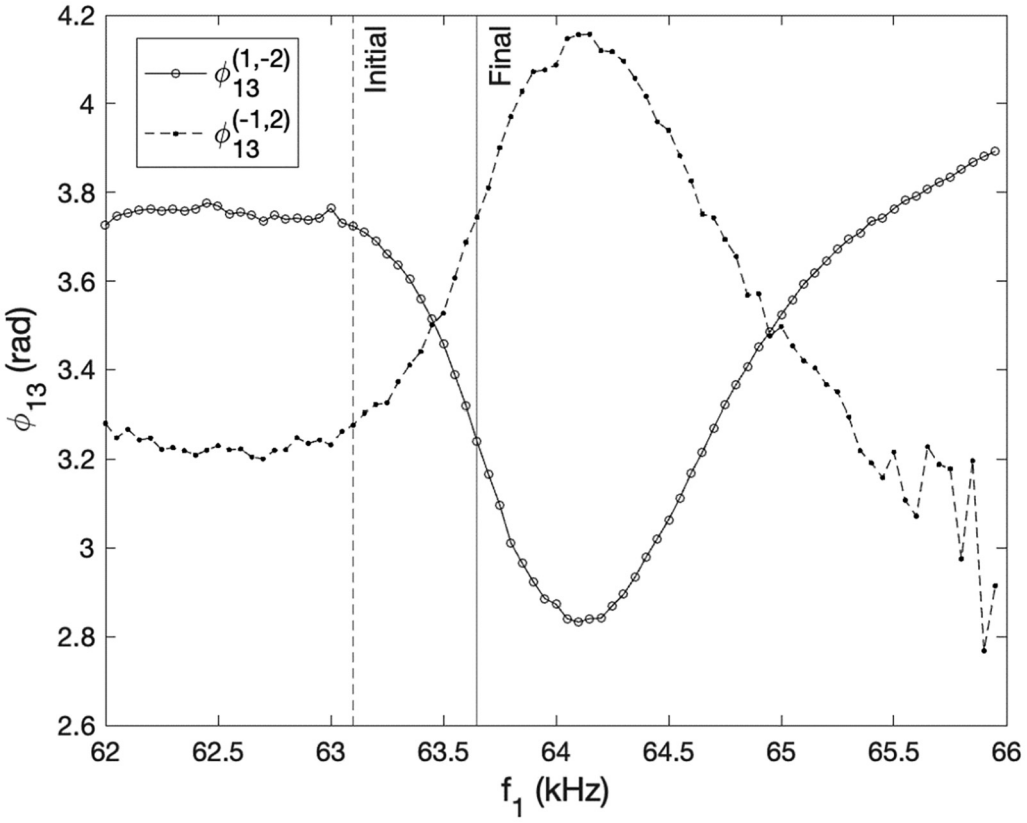

FIG. 8. Experimentally measured ϕ_{13} of phi-bit 1 ($\phi_{13}^{(1,-2)}$) and phi-bit 2 ($\phi_{13}^{(-1,2)}$) for implementing the permutation of components 2 and 4. Vertical lines (dashed and solid) as in Fig. 3.

transformation using the matrix below,

$$V^* = \begin{pmatrix} 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1C_2 \\ C_1S_2 \end{pmatrix} = \begin{pmatrix} C_1S_2 \\ S_1S_2 \\ S_1C_2 \\ C_1C_2 \end{pmatrix}. \quad (43)$$

Disregarding the phase change, one observed that we have successfully implemented a downward shift permutation of the state vector. The unitary transformation in Eq. (43) is effectively the product of a phase gate with the desired permutation gate as follows:

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

A cyclic permutation is implemented by expressing $\beta^{(i)}$, $i = 1, 2$ as a basic function of $\phi_{13}^{(p,q)}$ only by taking $\beta^{(i)} = \phi_{13}^{(p(i),q(i))}$ and using experimentally measured data of phi-bit 1 ($\phi_{13}^{(1,-1)}$) and phi-bit 2 ($\phi_{13}^{(-1,2)}$), shown in Fig. 9. Phi-bit 1 exhibits the behavior of a π-jump; however, the cyclic permutation requires a $\frac{\pi}{2}$ jump. To achieve this permutation, the experimental phase difference for phi-bit 1 was translated with a constant $\varepsilon^{(1)}$ and scaled the π jump to $\frac{\pi}{2}$ jump by a factor of 0.5. The experimental phase difference for phi-bit 2 was translated with a constant $\varepsilon^{(2)}$ such that the value lies between $\frac{\pi}{2}$ and $\frac{3\pi}{2}$. This translation is of the form $\beta^{(i)} = \phi_{13}^{(p(i),q(i))} + \varepsilon^{(i)}$, where $i = 1, 2$. The permutation is achieved by initializing the phi-bits at frequency $f_1 = 62.40$ kHz and tuning the frequency to 63.70 kHz. At the initial frequency, the initial ϕ_{13} for phi-bit 1 is 2.32 rad, and the initial ϕ_{13} for phi-bit 2 is 3.63 rad. During this tuning, the phase difference for phi-bit 1 and phi-bit 2 is exchanged such that the final phase difference for phi-bit 2 is the initial phase difference for phi-bit 1 plus $\frac{\pi}{2}$ radians, and the final phase difference for phi-bit 1 is the initial phase difference for phi-bit 2. The initial and final state vectors are shown below where we observed the components of the state vector have been shifted down,

$$\hat{V} = \begin{pmatrix} -0.34 \\ -0.65 \\ 0.60 \\ 0.32 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} 0.33 \\ 0.36 \\ 0.64 \\ 0.59 \end{pmatrix}. \quad (44)$$

The small deviations in the components of the initial and permuted state vectors are a result of experimental noise.

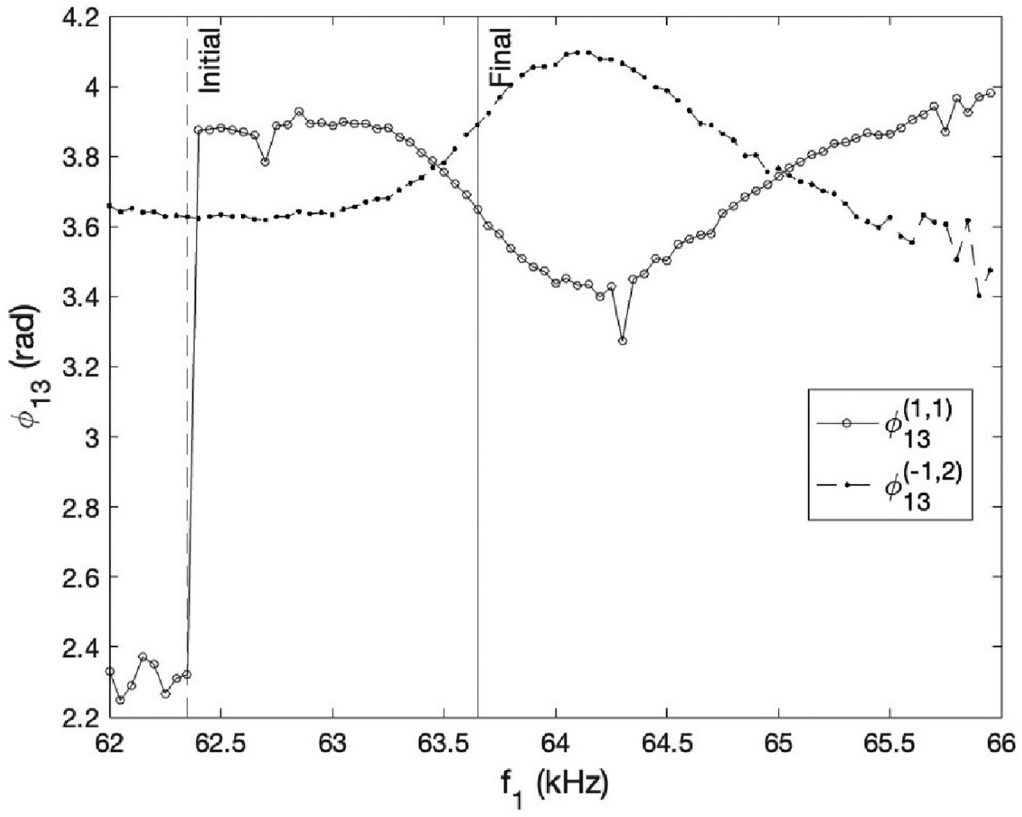

FIG. 9. Experimentally measured ϕ_{13} of phi-bit 1 ($\phi_{13}^{(1,1)}$) and phi-bit 2 ($\phi_{13}^{(-1,2)}$) for implementing a cyclic permutation. Vertical lines (dashed and solid) as in Fig. 3.

In Eq. (44), the initial vector, fixed input state, is determined by the constant functions $\beta^{(1)} = \phi_{13}^{(1,1)} + \varepsilon^{(1)} = 2.32$ and $\beta^{(2)} = \phi_{13}^{(-1,2)} + \varepsilon^{(2)} = 3.63$ at frequency $f_1 = 62.40$ kHz. However, one can apply this permutation to input states in the complete Hilbert space by using the same approach described in Sec. III B, which maps the desired input to the fixed input state.

Note that this cyclic permutation is effectively equivalent to performing a sequence of two different permutations. The first permutation is realized by the $\frac{\pi}{2}$ jump near 62.40 kHz. The second permutation is realized by the crossing in phases at the frequency 63.45 kHz. The sequence of these two permutations completes the cyclic permutation as follows:

$$\hat{V} = \begin{pmatrix} -0.34 \\ -0.65 \\ 0.60 \\ 0.32 \end{pmatrix} \rightarrow V' = \begin{pmatrix} 0.32 \\ 0.60 \\ 0.65 \\ 0.34 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} 0.33 \\ 0.36 \\ 0.64 \\ 0.59 \end{pmatrix}. \quad (45)$$

H. Summary of required initial phases

In Subsections III A–III G, we demonstrated the capability of realizing all possible permutations of the components of the state vector of a two phi-bits system with a single representation of that state vector. However, the theoretical and experimental realization of these permutations imposes some constraints on the range of values that can be spanned by the phi-bit phases $\beta^{(1)}$ and $\beta^{(2)}$. These constraints are summarized in Table I. These constraints are determined by the choice of representation. These constraints are imposed on the $\beta^{(i)}$; however, one has freedom in choosing the functions $\beta^{(i)}(\phi_{12}^{(i)}, \phi_{13}^{(i)})$, where $\phi_{12}^{(i)}$ and $\phi_{13}^{(i)}$ are determined by the experimental system and conditions.

TABLE I. Constraints imposed on the phi-bit phases required for specific permutation operation.

Permutation	$\beta^{(1)}$ (rad)	$\beta^{(2)}$ (rad)
SWAP	Any value	Any value
CNOT	$0 < \beta^{(1)} < \pi$	$\frac{-\pi}{2} < \beta^{(2)} < \frac{\pi}{2}$
Components 1 and 2	$0 < \beta^{(1)} < \pi$	$\frac{-\pi}{2} < \beta^{(2)} < \frac{\pi}{2}$
Components 1 and 3	$\pi < \beta^{(1)} < 2\pi$	$\frac{\pi}{2} < \beta^{(2)} < \frac{3\pi}{2}$
Components 1 and 4	Any value	Any value
Components 2 and 4	$\pi < \beta^{(1)} < 2\pi$	$\frac{\pi}{2} < \beta^{(2)} < \frac{3\pi}{2}$
Cyclic	$\frac{\pi}{2} < \beta^{(2)} < \frac{3\pi}{2}$	$\frac{\pi}{2} < \beta^{(2)} < \frac{3\pi}{2}$

Note also that all gates are exploiting the continuous behavior of phases in Table I but the SWAP gate (exploiting the sharp discrete jump behavior of phases) requires the mapping from any desired input to a fixed target input. The freedom one has in creating this mapping enables us to consider applying sequences of gates using different functions for the $\beta^{(1)}$ and $\beta^{(2)}$.

IV. SCALABLE INVERSION PERMUTATION

All the permutations implemented in Sec. III are realized with the very same representation of the two phi-bit state vectors [Eq. (10)]. Here, we employ an earlier representation of the state vector, that is, the tensor product shown in Eq. (9), to implement a permutation matrix $P = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$, which inverts all the elements of the state vector. This is to show that different permutations of the state vector of N phi-bits system can be achieved using an appropriate representation and we further demonstrate the viability of achieving a scalable permutation using the same representation.

Using the tensor product representation of two phi-bits described in Eq. (9), we are able to invert the entire state vector by taking the phase difference for each phi-bit through a $\pm\frac{\pi}{2}$ change,

$$V = \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1S_2 \\ C_1C_2 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} C_1C_2 \\ -C_1S_2 \\ -S_1C_2 \\ S_1S_2 \end{pmatrix}. \tag{46}$$

The minus signs arise from the trigonometric relation shown in Subsection III C. This inversion can be realized by initializing the two phi-bits state at arbitrary values and is equivalent to the unitary transformation matrix below,

$$V^* = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} S_1S_2 \\ S_1C_2 \\ C_1S_2 \\ C_1C_2 \end{pmatrix} = \begin{pmatrix} C_1C_2 \\ -C_1S_2 \\ -S_1C_2 \\ S_1S_2 \end{pmatrix}. \tag{47}$$

For experimental demonstration of this inversion, we start by expressing $\beta^{(i)}$, $i = 1, 2$ as a simple function of $\phi_{12}^{(p,q)}$ only by taking $\beta^{(i)} = \phi_{12}^{(p(i),q(i))}$ and using experimentally measured data of phi-bit 1 ($\phi_{12}^{(4,-3)}$) and phi-bit 2 ($\phi_{13}^{(4,-2)}$), shown in Fig. 10. We initialize the phi-bit at the frequency $f_1 = 62.10$ kHz and the effect of tuning the frequency to 62.55 kHz leads to a decrease in the phase difference of phi-bit 1 and 2 by $\frac{\pi}{2}$. The transformation of the state vector is shown below and disregarding the sign change in V^* one sees that the elements in the state vector are inverted,

$$V = \begin{pmatrix} -0.81 \\ -0.05 \\ -0.58 \\ -0.04 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -0.08 \\ 0.63 \\ 0.09 \\ -0.76 \end{pmatrix}. \tag{48}$$

Furthermore, we demonstrate the scalability of this operation by extending it to a system of three phi-bits. For this, we use a tensor product representation of the state vector.

The experimentally measured data of phi-bit 1 ($\phi_{12}^{(4,-3)}$), phi-bit 2 ($\phi_{12}^{(4,-2)}$), and phi-bit 3 ($\phi_{12}^{(4,-1)}$) are illustrated in Fig. 11. The frequency was again tuned from the same initial frequency of 62.10 kHz to the same final frequency of 62.55 kHz so that the phase difference for each phi-bit decreases by $\frac{\pi}{2}$ radians. The change in state vectors from this tuning is shown below,

$$V = \begin{pmatrix} S_1S_2S_3 \\ S_1S_2C_3 \\ S_1C_2S_3 \\ S_1C_2C_3 \\ C_1S_2S_3 \\ C_1S_2C_3 \\ C_1C_2S_3 \\ C_1C_2C_3 \end{pmatrix} = \begin{pmatrix} -0.76 \\ 0.28 \\ -0.05 \\ 0.02 \\ -0.54 \\ 0.20 \\ -0.03 \\ 0.01 \end{pmatrix} \rightarrow V^* = \begin{pmatrix} -C_1C_2C_3 \\ C_1C_2S_3 \\ C_1S_2C_3 \\ -C_1S_2S_3 \\ S_1C_2C_3 \\ -S_1C_2S_3 \\ -S_1S_2C_3 \\ S_1S_2S_3 \end{pmatrix} = \begin{pmatrix} -0.02 \\ -0.07 \\ 0.20 \\ 0.60 \\ 0.03 \\ 0.01 \\ -0.24 \\ -0.73 \end{pmatrix}. \tag{49}$$

Independently of some sign changes in Eq. (49) that arises from the trigonometric relation shown in Subsection III C, we have successfully inverted all the components of this state vector.

In Eqs. (48) and (49), the value of the component of the initial vector is determined by the functions $\beta^{(1)} = \phi_{12}^{(1,-2)}$ and $\beta^{(2)} = \phi_{12}^{(-1,2)}$ and the specific value of the phases at the frequency $f_1 = 63.45$ kHz. However, one can span the complete unit circle of phi-bit 1 and phi-bit 2 by using the same functions described in Sec. III A [Eqs. (18) and (19)], which translate the $\beta^{(1)}$ and $\beta^{(2)}$.

Note that the three phi-bits have the same $p = 4$ and we could use any other phi-bits by scaling. By this, we achieve a general scalable inversion for any number of phi-bits. Recall that logical phi-bits are nonlinear modes supported by the physical domain of the metamaterial (a volume of approximately 3×60 cm^3). Phi-bits do not have a specific size. Increasing the number of phi-bits does not increase the spatial extent of the physical system. The metamaterial can support tens to hundreds of logical phi-bits providing a basis for large scale operations. However, scaling to large numbers of phi-bits would require a tighter control on the driving frequency as well as on the sources of noise affecting the phi-bit phases ϕ_{12} and ϕ_{13}. Here, we have limited our demonstration to driving frequency increments of 50 Hz. An easily achievable tighter control of the frequency, with increments of 1 Hz, would significantly reduce error in the permuted elements of multi-phi-bit state vectors. Small uncertainties in ϕ_{12} and ϕ_{13} constitute another possible source of errors in operating the permutations. The magnitude of these uncertainties appears to be dependent on the amplitude of the

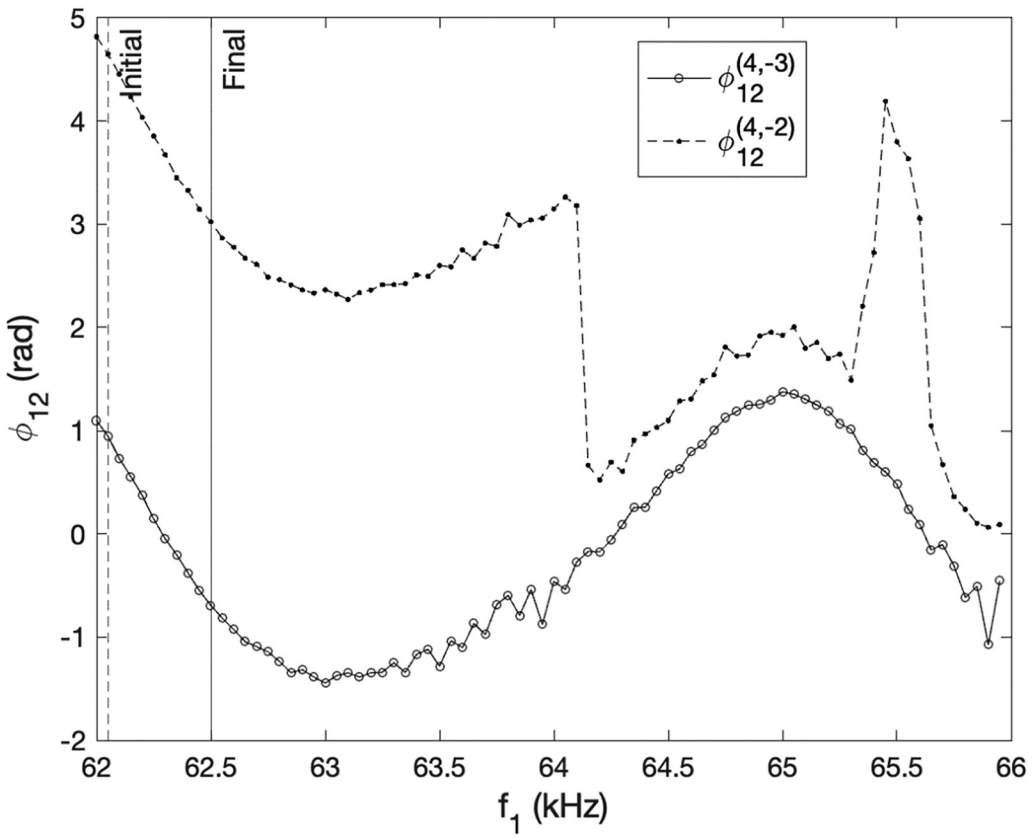

FIG. 10. Experimentally measured ϕ_{12} of phi-bit 1 ($\phi_{12}^{(4,-3)}$) and phi-bit 2 ($\phi_{12}^{(4,-2)}$) for implementing a 2 phi-bit inversion. Vertical lines (dashed and solid) as in Fig. 3.

nonlinear phi-bit mode, which varies with the driving frequency. Nonlinear modes with low amplitudes appear to exhibit larger uncertainties in phase. Considering a fluctuation in phase with magnitude $\Delta\phi$ rad, the permuted components of a N multi-phi-bit state vector may deviate to first order from their original values by the linear function $N\Delta\phi$. These deviations could become large for large numbers of phi-bits. The inversion permutation utilizes the continuous behavior of the phases. Since N phi-bits exhibit the same overall scaled continuous behavior, an averaging scheme could be used as a correction to smooth the data and significantly reduce that source of error. Using this approach of averaging with five phi-bits exhibiting continuous behavior, we were able to estimate a single phi-bit $\Delta\phi \sim \frac{1}{1000}$. This estimate is made by comparing the average to the linear combination of phases of the primary frequency $p\phi_{12}(f_1) + q\phi_{12}(f_2)$ and $p\phi_{13}(f_1) + q\phi_{13}(f_2)$. Indeed, amplitudes of the primary frequency are large, which have negligible uncertainty in the phases. This magnitude of deviation would enable permutation of the components of an $N = 50$ phi-bit state vector (i.e., 2^{50} components) with an accuracy of approximately $\frac{5}{100}$. However, averaging the phases of a large number of phi-bits is likely to decrease $\Delta\phi$ further. Additional reduction in phase variation can also be achieved by fabricating a metamaterial with a tighter tolerance than the one used here.

It is possible to achieve a permutation of multi-phi-bit state vectors with complex components. For instance, modifying the representation of a single logical phi-bit "j" given in Eq. (3) according to

$$V^{(j)} = \begin{pmatrix} e^{i\theta^{(j)}}\sin(\beta^{(j)}) \\ e^{-i\theta^{(j)}}\cos(\beta^{(j)}) \end{pmatrix}, \tag{50}$$

where $\theta^{(j)}$ is some function of the phi-bit phases.

The tensor product states of phi-bits 1 and 2 then take the form

$$V = \begin{pmatrix} e^{i(\theta^{(1)}+\theta^{(2)})}S_1S_2 \\ e^{i(\theta^{(1)}-\theta^{(2)})}S_1C_2 \\ e^{i(-\theta^{(1)}+\theta^{(2)})}C_1S_2 \\ e^{-i(\theta^{(1)}+\theta^{(2)})}C_1C_2 \end{pmatrix}. \tag{51}$$

As done in this section, we can operate on the S_j and C_j for each phi-bit through the β_j (which is effectively dependent on ϕ_{12}) to permute the product of trigonometric functions. To achieve the permutation of the complex components, we need a simultaneous change of sign of the $\theta^{(j)}$. This can be achieved by making $\theta^{(j)}$ a

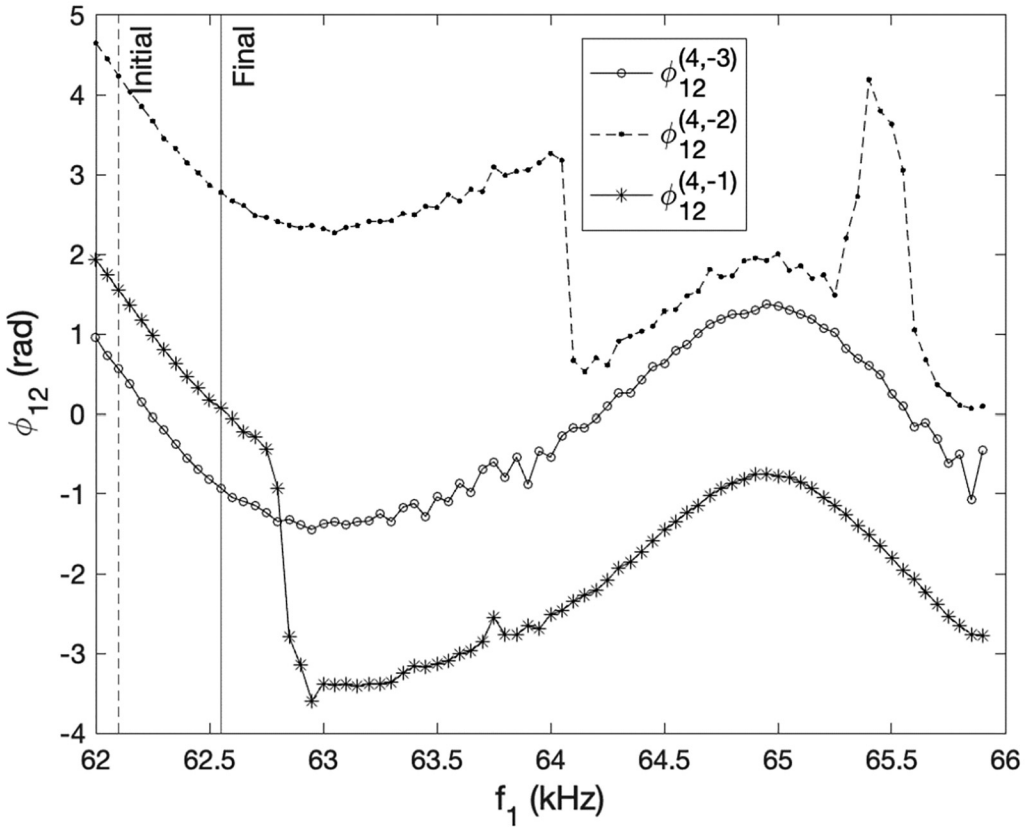

FIG. 11. Experimentally measured ϕ_{12} of phi-bit 1 ($\phi_{12}^{(4,-3)}$), phi-bit 2 ($\phi_{12}^{(4,-2)}$), and phi-bit 3 ($\phi_{12}^{(4,-1)}$), for demonstrating scalable inversion. Vertical lines (dashed and solid) as in Fig. 3.

function of the additionally available phase of the logical phi-bits, ϕ_{13}. This function can be designed to perform the desired sign change $\theta^{(j)}$ through changes of origin and scaling upon tuning the driving frequency.

As a final note concerning gates with $N \geq 3$ phi-bit, one can operate on only some of the phi-bits. The phi-bits that are not affected by a frequency shift are those exhibiting constants $\phi_{12}^{(p,q)}$ and $\phi_{13}^{(p,q)}$ over the corresponding range of driving frequency. By selecting phi-bits with varying and constant phases, one can operate on some phi-bits without affecting other phi-bits. For instance, in Fig. 9, phi-bit ($p = 1, q = 1$) has a $\phi_{13}^{(1,1)}$, which exhibit a $\frac{\pi}{2}$ jump between 62 and 63 kHz while phi-bit ($p = -1, q = 2$) and phi-bit ($p = 1, q = -2$) in Fig. 5 have both $\phi_{13}^{(-1,2)}$ and $\phi_{13}^{(1,-2)}$ remaining constant in that frequency range. A frequency shift from 62 to 63 kHz will, therefore, affect only the state of one phi-bit and not the other two.

V. QISKIT QUANTUM CIRCUIT COMPARISON

In this section, we compare some of the experimentally realized permutations in Secs. III and IV using phi-bits to their quantum counterparts using circuits developed with Qiskit. Qiskit is an open-source software development kit used in the design and optimization of quantum circuits.[23] We make use of the quantum simulator option of Qiskit using the available quantum gates in the Python library. Operations on quantum bits are realized with quantum circuits composed of sequential single and two qubit gates.[23] We begin with all qubits in the state $|0\rangle$, and for comparison purposes, we proceed to initializing the state vector of the qubit system to the same state vector as the phi-bit system. A quantum circuit breakdown into single and double qubit gates does not necessarily represent the number of physical actions required to achieve the quantum permutation operation. The implementation of gates in terms of physical actions depends on the quantum computing platform used so gate time could be a useful metric for comparing the efficiency of qubit-based and phi-bit-based permutation operations. However, for the sake of convenience, we utilize the number of physical actions necessary for quantum permutations as a metric to compare with phi-bit-based permutations achieved using a single physical action such as tuning the driving frequency f_1. In terms of operations time, a single phi-bit physical action is in the order of 1 ms. This is the time that is needed to establish a steady state for the acoustic wave in the waveguides. With a waveguide length of 0.6 m and longitudinal speed of sound of

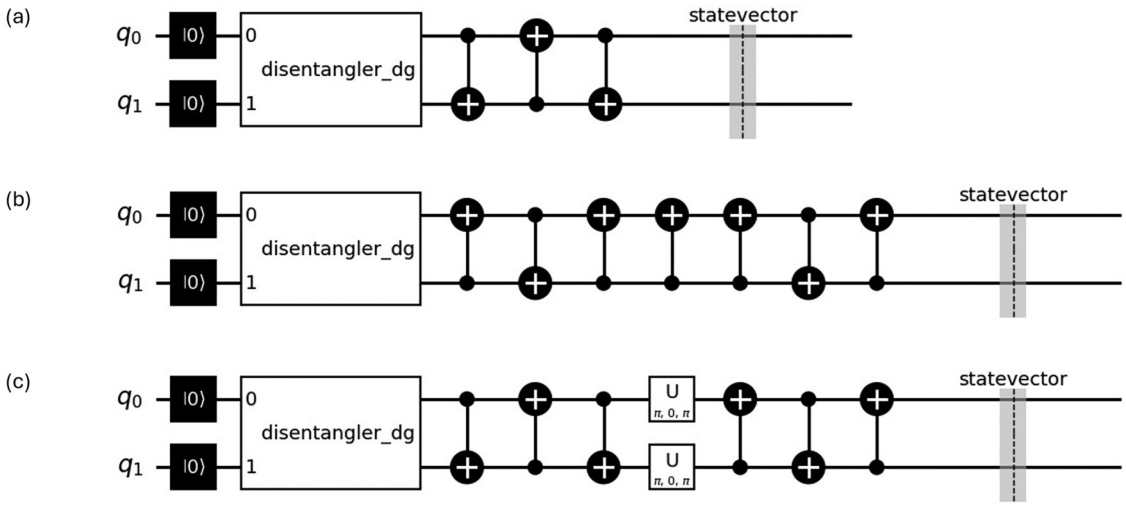

FIG. 12. Quantum circuit from Qiskit used to (a) achieve the SWAP gate and (b) permute components 2 and 4 of a two qubit state vector. (c) Quantum circuit to invert a two qubit state vector. The disentangler creates the initial state. The output state vector is indicated as a dashed vertical line.

approximately 5000 m/s, it takes 0.24 ms for a wave to travel back and forth along the waveguide. This time may not be favorable in terms of permutation with two phi-bits; however, a permutation of the 2^{50} components of an $N = 50$ phi-bit state vector would be realized in the same order of time. Furthermore, once at a steady state, the lifetime of phi-bits is arbitrarily long and determined by the steady state driving time of the system. Phi-bits can be maintained in a coherent state for an arbitrarily long time.

The SWAP gate is a defined quantum gate in Qiskit. The corresponding circuit is made up of three sequential CNOT gates as shown in Fig. 12(a). The circuit necessary to realize the permutation of components 2 and 4 of a two qubit state vector is illustrated in Fig. 12(b). The circuit requires seven sequential C-NOT gates. The Qiskit circuit for inverting a two qubit state vector uses a sequence of three C-NOT, two Pauli-X (single qubit), and two C-NOT gates as shown in Fig. 12(c).

Scaling the qubit-based inversion to a larger number of qubits will require an even more complicated circuit. Recall that the phi-bit-based operations require a single physical operation, illustrating the exceptional power of the phi-bit system. Due to the fragility of quantum states, to maintain data integrity for large scale qubit-based inversion may also require additional hardware (ancilla qubits) and/or hardware independent gates. Phi-bits are coherent and do not require such additional hardware and software.

VI. CONCLUSION

We describe a physical system constituted of an array of three externally driven acoustic waveguides. Logical phi-bits that are analogs of qubits appear as nonlinear modes in the system. The state of a logical phi-bit is represented on the Bloch sphere as a function of two phase differences measured between the waveguides. A modified tensor product representation of two logical phi-bit states with real components enables us to explore conditions leading to all possible permutations (including the SWAP and C-NOT gates) acting on all possible input states. Initialization of the input states and permutations are performed in separate stages. The initialization stage is performed in the linear scaling space of phases, while the permutation stage is performed in the product space of multiple phi-bits with the single physical action of changing the driving frequency. The permutation of the state vector components is achieved to within ± 1 factors associated with general phases of π. These permutations are demonstrated using experimental data from the driven system. One realizes the permutation unitary transformations by tuning the system driving frequency. The input is determined by the initial phi-bit phase difference and the associated representation. The phase differences are physically modified by tuning the frequency. The new phase differences and the initial representation determine the output. The unitary transformation is performed by physical action on the system itself. Additionally, we have introduced and demonstrated the scalability of the inversion of the components of a 2^N dimensional state vector arising from N logical phi-bits. We show that permutations are not limited to state vectors with real components but can also be realized with complex components. Changes in these complex components upon the single physical action can also be used to correct for the ± 1 factors.

We compare using logical phi-bits to their quantum counterparts in the context of the open-source software development kit, Qiskit. The result of the two approaches shows the advantage of logical phi-bits over equivalent quantum circuits. All Qiskit circuits employ a sequence of single and two qubit gates, unlike logical phi-bits that perform a simultaneous permutation of state vectors with just a single physical action.

In this paper, we limited the demonstration to single permutation operations, that is, we have not addressed sequences of operations on different sets (pairs) of phi-bits. However, sequences of operations can be achieved by tuning the frequency over

subsequent intervals. For instance, let us consider one pair of phi-bits, A and B, which have phases ϕ_{12} that exhibit continuous behavior and a sharp π jump at some frequency f, and another pair of phi-bits, C and D, which have phases ϕ_{12} that also exhibit continuous behavior and a π jump at a frequency $f' > f$. By subtracting the predicable continuous behavior from the phases of both pairs of phi-bits, we obtain ϕ'_{12} for phi-bits A, B, C, and D, which are essentially constant over all frequencies except for frequency f for the pair A and B and f' for pair C and D. Tuning the driving frequency such that it crosses f operates on some representation of the AB phi-bit state vector but not on the CD state vector. Further tuning of the frequency past f' operates on the CD state vector but not on the AB state vector. This is a simple illustration of how one can operate sequentially on two distinct pairs of phi-bits. More complex sequences of operations could be achieved while including the continuous behavior. This will be the subject of future work.

ACKNOWLEDGMENTS

This research was funded in part by the U.S. National Science Foundation (NSF) under Grant No. 2242925 through the Science and Technology Center New Frontiers of Sound (NewFoS). P.A.D., A.S.I., and D.C. also acknowledge partial support from NSF under Grant No. 2204400.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

All authors contributed equally toward the success of the research and writing of the manuscript.

David Cavalluzzi: Conceptualization (equal); Data curation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Akinsanmi S. Ige:** Data curation (equal); Formal analysis (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Keith Runge:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal). **Pierre A. Deymier:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (lead); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]Y. Ouyang, Y. Shen, and L. Chen, "Faster quantum computation with permutations and resonant couplings," Linear Algebra Appl. **592**, 270–286 (2020).
[2]K. J. Berry, J. E. Johnston, and P. W. Mielke, *A Primer of Permutation Statistical Methods* (Springer International Publishing, Cham, 2019).
[3]X. Wang, L. Feng, and H. Zhao, "Fast image encryption algorithm based on parallel computing system," Inf. Sci. **486**, 340–358 (2019).
[4]K. Berry, J. Johnston, and P. W. Mielke, Jr., *The Measurement of Correlation: A Permutation Statistical Approach* (Springer, 2018).
[5]C. A. Holt and S. P. Sullivan, "Permutation tests for experimental data," Exp. Econ. **26**(4), 775–812 (2023).
[6]C. Fu, J.-J. Chen, H. Zou, W.-H. Meng, Y.-F. Zhan, and Y.-W. Yu, *A Chaos-Based Digital Image Encryption Scheme with an Improved Diffusion Strategy References and Links* (Optica, 2012).
[7]D. R. I. M. Setiadi and N. Rijati, "An image encryption scheme combining 2D cascaded logistic map and permutation-substitution operations," Computation **11**(9), 178 (2023).
[8]M. Weber, "Quantum permutation matrices," Complex Anal. Oper. Theory **17**(3), 875–900 (2023).
[9]J. H. Reina, L. Quiroga, and N. F. Johnson, "Decoherence of quantum registers," Phys. Rev. A **65**(3), 032326 (2002).
[10]G. Li, Y. Ding, and Y. Xie, "Tackling the qubit mapping problem for NISQ-era quantum devices," in *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS* (Association for Computing Machinery, 2019), pp. 1001–1014.
[11]M. Moghaddaszadeh, M. Mousa, A. Aref, and M. Nouh, "Mechanical intelligence via fully reconfigurable elastic neuromorphic metasurfaces," APL Mater. **12**(5), 051117 (2024).
[12]S.-Y. Zuo, Y. Tian, Q. Wei, Y. Cheng, and X.-J. Liu, "Acoustic analog computing based on a reflective metasurface with decoupled modulation of phase and amplitude," J. Appl. Phys. **123**(9), 091704 (2018).
[13]S. Zuo, Q. Wei, Y. Tian, Y. Cheng, and X. Liu, "Acoustic analog computing system based on labyrinthine metasurfaces," Sci. Rep. **8**(1), 10103 (2018).
[14]S. Bringuier, N. Swinteck, J. O. Vasseur, J.-F. Robillard, K. Runge, K. Muralidharan, and P. A. Deymier, "Phase-controlling phononic crystals: Realization of acoustic Boolean logic gates," J. Acoust. Soc. Am. **130**(4), 1919–1925 (2011).
[15]S. A. R. Kuchibhatla and M. J. Leamy, "Topological insulator-based electro-acoustic transistors," in *Volume 12 35th Conference on Mechanical Vibration and Sound (VIB)*, (American Society of Mechanical Engineers, 2023).
[16]H. Pirie, S. Sadhuka, J. Wang, R. Andrei, and J. E. Hoffman, "Topological phononic logic," Phys. Rev. Lett. **128**(1), 015501 (2022).
[17]F. Li, P. Anzel, J. Yang, P. G. Kevrekidis, and C. Daraio, "Granular acoustic switches and logic elements," Nat. Commun. **5**, 5311 (2014).
[18]M. A. Hasan, K. Runge, and P. A. Deymier, "Experimental classical entanglement in a 16 acoustic qubit-analogue," Sci. Rep. **11**(1), 24248 (2021).
[19]K. Runge, P. A. Deymier, M. A. Hasan, T. D. Lata, and J. A. Levine, "Acoustic metamaterials for realizing a scalable multiple phi-bit unitary transformation," AIP Adv. **14**, 025010 (2024).
[20]C. M. Caves, C. A. Fuchs, and P. Rungta, "Entanglement of formation of an arbitrary state of two rebits," Found. Phys. Lett. **14**, 199–212 (2001).
[21]N. Prasannan, S. De, S. Barkhofen, B. Brecht, C. Silberhorn, and J. Sperling, "Experimental entanglement characterization of two-rebit states," Phys. Rev. A **103**(4), L040402 (2021).
[22]W. K. Wootters, "Entanglement sharing in real-vector-space quantum theory," Found. Phys. **42**(1), 19–28 (2012).
[23]A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," arXiv:2405.08810 (2024).