

All You Need is pi: Quantum Computing with Hermitian Gates

Ben Zindorf and Sougato Bose

Department of Physics and Astronomy, University College London, Gower Street, WC1E 6BT London, United Kingdom

Universal gate sets for quantum computation, when single and two qubit operations are accessible, include both Hermitian and non-Hermitian gates. Here we utilize the fact that any single-qubit operator may be implemented as two Hermitian gates, and thus a purely Hermitian universal set is possible. This implementation can be used to prepare high fidelity single-qubit states in the presence of amplitude errors, and helps to achieve a high fidelity single-qubit gate decomposition using four Hermitian gates. An implementational convenience can be that non-identity single-qubit Hermitian gates are equivalent to π rotations up to a global phase. We show that a gate set comprised of π rotations about two fixed axes, along with the CNOT gate, is universal for quantum computation. Moreover, we show that two π rotations can transform the axis of any multi-controlled unitary, a special case being a single CNOT sufficing for any controlled π rotation. These gates simplify the process of circuit compilation in view of their Hermitian nature. We exemplify by designing efficient circuits for a variety of controlled gates, and achieving a CNOT count reduction for the four-controlled Toffoli gate in LNN-restricted qubit connectivity.

1 Introduction

Pivotal requirements in the field of quantum computation are highly reliable gates, as well as efficient quantum circuits (i.e, a low number of fundamental gates being used to implement a given unitary operation). As it stands, universal gate sets for quantum computation, consisting of fundamental one- and two-qubit gates, are composed of both Hermitian gates, such as the CNOT, Hadamard, and Pauli gates, as well as Non-Hermitian gates, such as the S and T gates [1]. It is known that if you allow for three qubit gates, then Toffoli (which is Hermitian) allows for universal quantum computation along with the Hadamard. However, typically single and two qubit gates are used to implement the Toffoli. One can then ask a simple, but fundamental, question: can we do universal quan-

tum computation purely by using single and two qubit Hermitian gates? As the CNOT is already Hermitian, the question effectively becomes as to whether all the required single-qubit unitaries can be Hermitian. As we will discuss in this paper, the answer is affirmative. The usage of Hermitian single-qubit gates then immediately has some positive implications for the implementations of quantum computation.

Firstly, Hermitian single-qubit gates can be implemented, up to a global phase, as π rotations about any axis in the Bloch sphere, also called a π -pulse in experimental terminology [2]. It effectively means that one switches on a Hamiltonian $g\hat{\sigma}\cdot\hat{v}$, where g is a coupling strength, and $\hat{v}$ is the unit vector along any axis of the Bloch sphere, to act on the qubit for an amount of time $t = \pi/g$. For a reliable gate, the time t has to be controlled very accurately. Typically, this time is a duration for which a voltage/microwave/laser pulse is applied to the qubit. Thus, one can set a control machine to generate one such exquisite pulse of duration giving a π pulse to a very high precision, and then repeat the same pulse settings for *all* the gates, just by changing the axis $\hat{v}$ of the rotation in the Bloch sphere. It is plausible that in some technologies, the axes of rotation can be more precisely set than the time duration. If we have electromagnetic pulses of fixed duration, the axis of the Bloch sphere can be set to be anywhere in the $x-y$ plane by choosing their phase ϕ as $\hat{v} = (\cos \phi, -\sin \phi)$ [3]. Such rotations, along with Hadamard (which is also a π rotation, albeit about a different axis), suffice. Moreover, as we will show, we require *only two fixed* axes for a universal Hermitian gate set, so that in any technology, one can focus on setting them precisely. In addition, composite pulses have been shown to significantly improve the fidelity of a π pulse in the presence of pulse-strength errors (aka amplitude errors), while applying this to other angles is more challenging [4, 5, 6, 7]. These robust π pulses are a critical ingredient of dynamical decoupling [8, 9, 10, 11], which is used to protect a qubit from the environment and reduce its decoherence rate. For the same reason, π pulses have been extensively refined. For the task of suppressing amplitude errors, we provide a sequence of two π rotations which improves the fidelity of single-qubit state preparation. Similarly, we provide a high fidelity decomposition of arbitrary single-qubit gates using four π rotations in the presence of amplitude errors.

Ben Zindorf: ben.zindorf.19@ucl.ac.uk

Secondly, a positive implication stems from the inherent characteristic of Hermitian gates being their own inverse. This allows them to be used as a symbolic tool to simplify circuit optimization and compilation processes in many cases. As we will show, the Hermitian gates provide a useful framework, especially for the task of decomposing controlled gates. For example, it allows us to change the axis of rotation of any Multi-Controlled unitary to any other axis. This allows one to change the CNOT to any controlled π rotation. We utilize this framework to provide a decomposition of any controlled $U(4)$ gate using a small number of CNOT gates. The Hermitian gates can then be replaced with gates from any choice of universal set in an efficient manner. In addition, we use the same framework to find efficient decompositions of various types of multi-controlled single-target gates with a small number of controls. While some of our circuits require a CNOT count which matches the state-of-the-art, others provide generalized versions of these gates, with no increase of CNOT count. In the case of a Toffoli gate with four controls under the restrictions of LNN/chain connectivity, our methods resulted in a reduction of CNOT gates compared to the state-of-the-art [12].

Here we summarize the results of the paper. We start by providing a geometrically motivated description of Hermitian π -rotations and using this to demonstrate the implications of known mathematical concepts, such as Householder reflections [13], for single qubit rotations – demonstrating the ability to use two Hermitian gates to decompose any single-qubit operator, or to transform the rotation axis of a given one. We then present a variety of universal gate sets comprised of *restricted* subsets of all Hermitian gates, smallest of which requires only the ability to apply the CNOT gate, and to rotate the Bloch sphere about *two fixed axes* by the *fixed angle* π for universal quantum computation. We then present our π -rotation sequences which provide a quadratic suppression of amplitude errors, using four π -rotations for any single-qubit unitary, and only two π -rotations for state preparation. We then show that the single-qubit axis transformations can be applied to the target operator of controlled gates, which immediately provides improved decompositions of multi-controlled gates compared to state of the art. This also implies that one CNOT gate suffices to implement any singly-controlled π -rotation, while any other choice of angle requires two CNOT gates, providing an alternative geometrical description of Lemma 5.5 in [14]. We continue to show that Hermitian gates can be used as an intuitive symbolic tool for synthesizing multi-controlled gates, specifically resulting in a reduced CNOT count of the four-controlled Toffoli gate, compared to the state-of-the-art decompositions which relied on machine learning or exhaustive search. Finally we

demonstrate the ability to add a control to a given unitary using the same formalism, exemplifying on $U(4)$ gates.

2 Single qubit operators

We define a π -rotation as the Hermitian operator $\Pi(\hat{v}) := iR_{\hat{v}}(\pi)$, such that $R_{\hat{v}}(\pi) \in SU(2)$ applies a rotation by π about the axis $\hat{v} = (v_x, v_y, v_z)$. The axis can be represented by the standard spherical coordinates $\hat{v}(\theta, \phi) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$. A π -rotation about the axis $\hat{v}$ can thus be written in the two following ways:

$$\Pi(\hat{v}) = \begin{pmatrix} v_z & v_x - iv_y \\ v_x + iv_y & -v_z \end{pmatrix}$$

$$\Pi(\theta, \phi) := \Pi(\hat{v}(\theta, \phi)) = \begin{pmatrix} \cos \theta & e^{-i\phi} \sin \theta \\ e^{i\phi} \sin \theta & -\cos \theta \end{pmatrix}$$

Applying two π -rotations about the *same axis* results in a 2π -rotation, and thus $(\Pi(\hat{v}))^2 = -R_{\hat{v}}(2\pi) = I$. Interestingly, any $SU(2)$ operator can be implemented as two π -rotations, as we state in Lemma 1 and prove in Appendix B. Since $SU(2)$ is a double cover of $SO(3)$, this Lemma can also be inferred from [15, 16] in the mathematics literature, however, its implications for quantum computing seem to have been overlooked. We mark $\angle_{\hat{v}_2}^{\hat{v}_1}$ as the angle between two unit vectors $\hat{v}_1, \hat{v}_2$, and $\hat{R}_{\hat{v}}(\theta)$ as the three dimensional rotation matrix by angle θ about the axis $\hat{v}$.

Lemma 1. *Any $R_{\hat{v}}(\lambda) \in SU(2)$ operator can be implemented as $\Pi(\hat{v}_2)\Pi(\hat{v}_1)$ with $\hat{v}_1$ as any unit vector perpendicular to $\hat{v}$, and $\hat{v}_2 = \hat{R}_{\hat{v}}(\frac{\lambda}{2})\hat{v}_1$ with $\frac{\lambda}{2} \in (-\pi, \pi]$.*

As $U(2)$ operators can be implemented as $SU(2)$ up to a global phase, Lemma 1 shows that any single qubit operator can be implemented as two π -rotations (Figure 1a). This implementation requires four parameters - two for each Π gate, while $SU(2)$ gates require only three. However, one of the vectors $\hat{v}_1, \hat{v}_2$ can always be chosen on the xy plane, so that it is perpendicular to both the $\hat{z}$ axis and the vector $\hat{v}$. Therefore, one of the π -rotations only requires one parameter, since it can be described as $\Pi(\frac{\pi}{2}, \phi)$. Moreover, since $\Pi(-\hat{v}) = -\Pi(\hat{v})$ for any $\hat{v}$, the required π -rotations can always be chosen from one half of the Bloch sphere, e.g. the angles defining $\Pi(\theta, \phi)$ can be constrained to $\theta, \phi \in [0, \pi)$, potentially introducing a -1 global phase to the $SU(2)$ gate.

Arbitrary rotations about the $\hat{x}, \hat{y}$, and $\hat{z}$ axes require a single parameter. In the π -rotation decomposition, this can be achieved by choosing one of the vectors $\hat{v}_1, \hat{v}_2$ to be in a fixed position. For example, a rotation about $\hat{z}$ requires two vectors on the xy plane, and thus a comfortable choice would be the $\hat{x}$ axis

Single qubit π -rotations can be used to transform the rotation axis of any given single-qubit gate to any other axis, as we show next. We define the set of unit vectors $M(\hat{v}_1, \hat{v}_2)$ for any $\hat{v}_1, \hat{v}_2$, such that $\hat{v}_M \in M(\hat{v}_1, \hat{v}_2)$ iff $\hat{v}_2 = \hat{R}_{\hat{v}_M}(\pi)\hat{v}_1$. In case $\hat{v}_1 = -\hat{v}_2$, the set $M(\hat{v}_1, \hat{v}_2)$ holds all of the vectors perpendicular to $\hat{v}_1$, and otherwise, the set holds the vector $\hat{v}_M = \frac{\hat{v}_1 + \hat{v}_2}{|\hat{v}_1 + \hat{v}_2|}$ which is in the middle between $\hat{v}_1$ and $\hat{v}_2$. Therefore, $M(\hat{v}_1, \hat{v}_2) \neq \emptyset$ for any choice of $\hat{v}_1, \hat{v}_2$.

Proof. Immediate from Lemma 1, noting that $\Pi(\hat{v}')\Pi(\hat{v}_M)$ and $\Pi(\hat{v}_M)\Pi(\hat{v})$ implement the same rotation. $\square$

Lemma 3. $R_{\hat{v}}(\lambda) = \Pi(\hat{v}_M)R_{\hat{v}'}(\lambda)\Pi(\hat{v}_M)$ for any angle λ , unit vectors $\hat{v}, \hat{v}'$, and $\hat{v}_M \in M(\hat{v}, \hat{v}')$.

The Hermitian CNOT gate, along with arbitrary single-qubit operators are universal for quantum computation. Since any single qubit operator can be implemented using two π -rotations as per Lemma 1, the Hermitian set $\{CNOT, \Pi(\theta, \phi)\}$ with $\theta, \phi \in [0, \pi)$ is universal as well. Similarly, the universality of $\{CNOT, H, R_{\hat{z}}\}$ provides the Hermitian universal set $\{CNOT, H, \Pi(\frac{\pi}{2}, \phi)\}$ with $\phi \in [0, \frac{\pi}{2}]$. The latter Hermitian set requires significantly less available axes for π -rotations than the former. Since $S := P(\frac{\pi}{2}) = \Pi_S X$, and $T := P(\frac{\pi}{4}) = \Pi_T X$ up to a global phase, with $\Pi_S := \Pi(\frac{\pi}{2}, \frac{\pi}{4})$, $\Pi_T := \Pi(\frac{\pi}{2}, \frac{\pi}{8})$ and $P(\psi) := e^{i\frac{\psi}{2} R_{\hat{z}}}(\psi)$ as the phase gate, the Hermitian sets

$$\begin{array}{c} q: \\ a: \end{array} \begin{array}{c} \oplus \\ \bullet \end{array} \Pi_T \begin{array}{c} \oplus \\ \bullet \end{array} \Pi_T = \begin{array}{c} q: \\ a: \end{array} \begin{array}{c} \oplus \\ \bullet \end{array} \boxed{H} \bullet \boxed{H} \begin{array}{c} \oplus \\ \bullet \end{array} \boxed{H} \bullet \boxed{H}$$

Group	Standard Set	Hermitian Set	Range
Universal	$\{CNOT, U\}$ $\{CNOT, H, R_z\}$ $\{CNOT, H, T\}$ $\{CNOT, H, T\}$	$\{CNOT, \Pi(\theta, \phi)\}$ $\{CNOT, H, \Pi(\frac{\pi}{2}, \phi)\}$ $\{CNOT, H, \Pi_T, X\}$ $\{CNOT, H, \Pi_T\}$	$\theta, \phi \in [0, \pi)$ $\phi \in [0, \frac{\pi}{2}]$
Clifford	$\{CNOT, H, S\}$ $\{CNOT, H, S\}$	$\{CNOT, H, \Pi_S, X\}$ $\{CNOT, H, \Pi_S\}$	

4 Amplitude error suppression

Accepted in *Quantum* 2025-11-16, click title to verify. Published under CC-BY 4.0.

to $\hat{v}$ to implement two instances of $R_{\hat{v}}(\frac{\lambda}{2})$ in different ways. An additional benefit unique to Hermitian π -rotation is the freedom to replace a $\Pi(\hat{v}_j)$ gate with $\Pi(-\hat{v}_j)$ while only introducing a global phase of -1 . Utilizing these freedoms for the task of minimizing the infidelity has resulted in the following.

We use Lemma 1 to decompose $R_{\hat{v}}(\lambda)$ using four π -rotations as $\Pi_4\Pi_3\Pi_2\Pi_1$ such that $\Pi_j = \Pi(\hat{v}_j)$ and $\Pi_4\Pi_3 = \Pi_2\Pi_1 = -R_{\hat{v}}(\frac{\lambda}{2})$. The axis $\hat{v}_1$ is perpendicular to $\hat{v}$, and the rest are defined as $\hat{v}_{j+1} = \hat{R}_{\hat{v}}(\frac{\lambda}{4} + j\pi)\hat{v}_j$ (Figure 1b). In the presence of amplitude errors, each Π_j gate is replaced by $\Pi_j R_{\hat{v}_j}(\epsilon\pi)$. While we had the freedom to choose $\hat{v}_3$ as any vector perpendicular to $\hat{v}$, our specific choice is beneficial, as it provides the following commutation rules which can be obtained using Lemma 3: $R_{\hat{v}_4}(\epsilon\pi)\Pi_3 = \Pi_3 R_{-\hat{v}_2}(\epsilon\pi)$ and $\Pi_2 R_{\hat{v}_1}(\epsilon\pi) = R_{-\hat{v}_3}(\epsilon\pi)\Pi_2$. The four imperfect Π gates decomposition can therefore be written as $\Pi_4\Pi_3 E \Pi_2 \Pi_1 = R_{\hat{v}}(\frac{\lambda}{2}) E R_{\hat{v}}(\frac{\lambda}{2})$ with

$$E := R_{-\hat{v}_2}(\epsilon\pi) R_{\hat{v}_3}(\epsilon\pi) R_{\hat{v}_2}(\epsilon\pi) R_{-\hat{v}_3}(\epsilon\pi),$$

and the infidelity is given by $\mathcal{I} = 1 - |\text{tr}(E)|/2$. From the definition of $\hat{v}_2$ and $\hat{v}_3$, and since the trace is invariant under a change of basis, we can solve for $\hat{v} = \hat{z}$, $\hat{v}_2 = \hat{y} = v(\frac{\pi}{2}, \frac{\pi}{2})$ and $\hat{v}_3 = v(\frac{\pi}{2}, \frac{\pi}{2} + \frac{\lambda}{4})$. Using Lemma 1 we can write for this choice $R_{\hat{v}_2}(\epsilon\pi) = \Pi(\frac{\pi\epsilon}{2}, 0)Z$ and $R_{\hat{v}_3}(\epsilon\pi) = \Pi(\frac{\pi\epsilon}{2}, \frac{\lambda}{4})Z$. Setting $a = \cos^2(\frac{\pi\epsilon}{2}) + e^{i\frac{\lambda}{4}} \sin^2(\frac{\pi\epsilon}{2})$ and $b = (e^{-i\frac{\lambda}{4}} - 1) \cos(\frac{\pi\epsilon}{2}) \sin(\frac{\pi\epsilon}{2})$ provides the following from the definition of $\Pi(\theta, \phi)$

$$\begin{aligned} \text{tr}(E) &= \text{tr}((Z\Pi(\frac{\pi\epsilon}{2}, 0)\Pi(\frac{\pi\epsilon}{2}, \frac{\lambda}{4}))^2) = \\ \text{tr} \left(\begin{pmatrix} a & b \\ b^* & -a^* \end{pmatrix}^2 \right) &= a^2 + a^{*2} + 2b^*b = \\ 2\cos^2(\frac{\pi\epsilon}{2}) (\cos^2(\frac{\pi\epsilon}{2}) + 2\sin^2(\frac{\pi\epsilon}{2})) &+ 2\cos(2\frac{\lambda}{4}) \sin^4(\frac{\pi\epsilon}{2}). \end{aligned}$$

Substituting $\cos^2(\frac{\pi\epsilon}{2}) = 1 - \sin^2(\frac{\pi\epsilon}{2})$ and $\cos(2\frac{\lambda}{4}) = 1 - 2\sin^2(\frac{\lambda}{4})$ provides $\text{tr}(E) = 2 - 4\sin^4(\frac{\pi\epsilon}{2})\sin^2(\frac{\lambda}{4})$. We get $|\text{tr}(E)| = \text{tr}(E)$ when the amplitude error is sufficiently small ($|\epsilon| < 0.6$), and therefore, by definition, $\mathcal{I} = 2\sin^4(\frac{\pi\epsilon}{2})\sin^2(\frac{\lambda}{4}) \approx \epsilon^4 \frac{\pi^4}{8} \sin^2(\frac{\lambda}{4})$. This provides a quadratic improvement compared to the infidelity of the direct implementation $R_{\hat{v}}((1+\epsilon)\lambda)$ given by $1 - \cos(\frac{\epsilon\lambda}{2}) \approx \epsilon^2 \lambda^2/8$. We provide a numerical analysis in Figure 2, comparing these implementations with the Euler ZYZ, and the two Π gate (from Lemma 1 with $\hat{v}_2 \Rightarrow -\hat{v}_2$) decompositions. As can be seen, the latter two decompositions result in a similar infidelity for the chosen rotations.

Although other four-step decompositions have been proposed [19, 20, 21] in the context of composite pulses, achieving similar benefits, we are not aware of such a decomposition which achieves *arbitrary* rotations using four π rotations/pulses.

By relaxing the requirement to implement a specific rotation, we now focus on transforming an arbitrary initial single-qubit state $|\Psi_A\rangle$ to an arbitrary

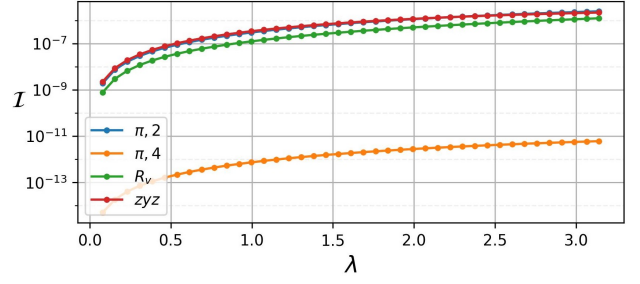


Figure 2: Numerical gate infidelity (log scale) as a function of the rotation angle λ with a fixed axis $\hat{v}(\frac{\pi}{2}, \frac{\pi}{16})$ and an amplitude error $\epsilon = 10^{-3}$. Comparing the two Π , four Π , Euler ZYZ and the direct R_v implementations.

final state $|\Psi_B\rangle$ with the corresponding state vectors $\hat{v}_A, \hat{v}_B$ respectively. It has previously been shown that a single π rotation about the axis $\frac{\hat{v}_A + \hat{v}_B}{|\hat{v}_A + \hat{v}_B|}$ suffices for this task [22]. Moreover, since π is the only angle which allows to transform $|0\rangle$ to $|1\rangle$ in one step, it immediately follows that π is the *only* rotation angle which allows to transform *any* initial state to *any* final state. Generally, many trajectories can be chosen between $\hat{v}_A$ and $\hat{v}_B$, for example, a geodesic trajectory can be achieved by a $R_{\hat{v}_\perp}(\lambda)$ rotation with $\hat{v}_\perp$ perpendicular to both $\hat{v}_A, \hat{v}_B$ and $\lambda = \angle_{\hat{v}_\perp}^{\hat{v}_A, \hat{v}_B}$. In the presence of amplitude errors, we find that using the two Π gate decomposition from Lemma 1 can provide a quadratic improvement to the fidelity of the final state, compared to the geodesic approach. The state infidelity can be written as $\mathcal{I}_{|\rangle} = 1 - |\langle\Psi_B|\Psi'_B\rangle|^2$, with $|\Psi'_B\rangle$ as the final state achieved by imperfect rotations. We decompose $R_{\hat{v}_\perp}(\lambda)$ as $\Pi_2\Pi_1$ using Lemma 1 with $\hat{v}_2 \Rightarrow -\hat{v}_2$ as before, while taking advantage of the fact that $\hat{v}_1$ can be chosen as any vector perpendicular to $\hat{v}_\perp$. We find that the ideal choice in this case is $\hat{v}_1 = \hat{R}_{\hat{v}_\perp}(\frac{\pi}{2} + \frac{\lambda}{4})\hat{v}_A$. As can be seen in Figure 3, this provides a quadratic improvement compared to the geodesic trajectory, the Euler ZYZ, and the single Π decompositions.

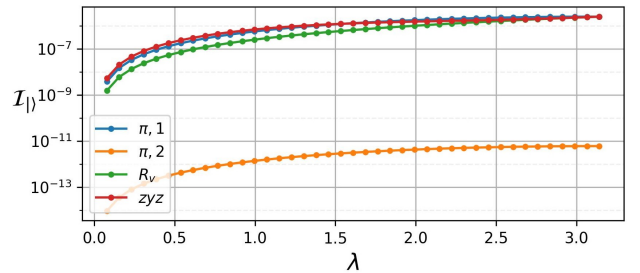


Figure 3: Numerical state infidelity (log scale) as a function of λ , transforming the $|0\rangle$ state to a final state defined by the state vector $\hat{v}_B = \hat{v}(\lambda, \frac{\pi}{2})$. Comparing the single Π , two Π , Euler ZYZ and the geodesic ($R_{v_\perp}$) implementations with amplitude error $\epsilon = 10^{-3}$.

5 Controlled Π Gates

In this section we show that a single-controlled $U \in U(2)$ gate can be implemented using one *CNOT* gate iff $U = e^{i\psi}\Pi(\hat{v})$ for any unit vector $\hat{v}$ and any angle ψ . Furthermore, we provide a method to transform the rotation axis of any multi-controlled $U(2)$ gate to any other axis using two single qubit π -rotations.

We use $[\cdot]_t^C$ to mark a multi-controlled gate with a control set C holding n qubits, and a target qubit t . Similarly, $[\cdot]_t$ marks a single qubit gate applied on t .

Lemma 4. $[\omega R_{\hat{v}}(\lambda)]_t^C = [\Pi(\hat{v}_M)]_t [\omega R_{\hat{v}'}(\lambda)]_t^C [\Pi(\hat{v}_M)]_t$ with $\omega := e^{i\psi}$ for any angles λ, ψ , unit vectors $\hat{v}, \hat{v}'$, and $\hat{v}_M \in M(\hat{v}, \hat{v}')$.

Proof. The gate $[\omega R_{\hat{v}'}(\lambda)]_t^C$ applies $\omega R_{\hat{v}'}(\lambda)$ on qubit t if set C is in state $|11..1\rangle$ ("on"), and I otherwise. For $[\Pi(\hat{v}_M)]_t [\omega R_{\hat{v}'}(\lambda)]_t^C [\Pi(\hat{v}_M)]_t$, if set C is "on", we get $\omega \Pi(\hat{v}_M) R_{\hat{v}'}(\lambda) \Pi(\hat{v}_M) = \omega R_{\hat{v}}(\lambda)$ from Lemma 3, and $\Pi(\hat{v}_M) \Pi(\hat{v}_M) = I$ otherwise. $\square$

Existing work which addresses low CNOT count implementations of multi-controlled gates largely distinguish between the most general form of these gates, allowing for any rotation angle λ about any axis $\hat{v}$, and the less general form, allowing any angle λ about a fixed axis $\hat{v}'$. Focusing on the latter allows to simplify the problem, which may result in a lower gate count compared to the former. However, Lemma 4 suggests that this lower gate count is applicable to both, at the cost of two additional single-qubit Π gates. These Π gates are applied on both sides of a given implementation of the multi-controlled gate, without requiring any further adjustments. For example, the authors of [23] show that any multi-controlled $SU(2)$ gate can be implemented using $20n + O(1)$ CNOT gates, while only $16n + O(1)$ CNOT gates are required if the rotation is about the $\hat{v}' = z$ axis. Lemma 4 immediately implies that any multi-controlled $SU(2)$ gate can be implemented using $16n + O(1)$ CNOT gates – a 20% improvement over the state-of-the-art. We note that the axis transformation from Lemma 4 is not restricted to multi-controlled $SU(2)$ gates, but depending on the choice of ψ , it can be applied to multi-controlled $U(2)$ as well. This also allows to use well studied implementations of the multi-controlled Pauli-X gate to implement a multi-controlled Hadamard with the same CNOT count, by simply noting that both X and H are Hermitian π -rotations, albeit about a different axis. For example, this would allow the results of [24] to be modified, using our Lemma 4, resulting in a multi-controlled Hadamard cost of $12n$ instead of $32n$ – an immediate 62.5% reduction (which was already known from [25] with a more relaxed ancilla requirement).

For the case of a single control, by setting $\lambda = \pi, \hat{v}' = \hat{x}, C = \{c\}$, and replacing ψ with $\psi + \frac{\pi}{2}$, Lemma 4 shows that any $[e^{i\psi}\Pi(\hat{v})]_t^c$ gate (e.g. controlled Hadamard) can be implemented using a single

CNOT gate simply by choosing $\hat{v}_M \in M(\hat{v}, \hat{x})$ as:

$$\begin{array}{c} c : \text{---} \bullet \text{---} \\ t : \text{---} [e^{i\psi}\Pi(\hat{v})] \text{---} \end{array} = \begin{array}{c} c : \text{---} [P(\psi)] \text{---} \bullet \text{---} \\ t : \text{---} [\Pi(\hat{v}_M)] \text{---} \oplus \text{---} [\Pi(\hat{v}_M)] \text{---} \end{array} \quad (1)$$

In Lemma 12 (Appendix B), we show that these controlled π -rotations are the *only* controlled gates which can be implemented using a single CNOT gate. Combining this, and Lemma 4 for a single control, proves the following theorem. This theorem can also be realized from Lemma 5.5 in [14].

Theorem 1. A single-controlled unitary gate $[U]_t^c, U \in U(2)$ can be implemented using one CNOT gate iff $U = e^{i\psi}\Pi(\hat{v})$ for any unit vector $\hat{v}$ and any angle ψ .

This realization provides sufficient motivation to use controlled π -rotations as elementary gates when constructing large circuits, especially if the goal is to minimize the count of CNOT gates. We demonstrate this in the next section.

6 Controlled $SU(2)/U(2)$ gates

In this section we utilize the Hermitian gate framework in order to implement controlled gates using a small CNOT count. We emphasize the simplicity in which our structures are developed, while reproducing, generalizing or improving state-of-the-art methods. Starting with a controlled $W \in SU(2)$ (CW), we can implement $W = R_{\hat{v}}(\lambda)$ as $\Pi_2 \Pi_1$ with $\hat{v}_1, \hat{v}_2$ perpendicular to $\hat{v}$, and $\angle_{\hat{v}_2}^{\hat{v}_1} = \frac{\lambda}{2}$ using Lemma 1. Adding a control to each of the Π gates only requires a single CNOT according to Theorem 1. Circuit (2) therefore requires two CNOT gates, similarly to the well known decomposition from [14]. Clearly, any controlled $U \in U(2)$ (CU) gate can be implemented by applying a phase gate on the control qubit.

$$\begin{array}{c} c : \text{---} \bullet \text{---} \\ t : \text{---} [W] \text{---} \end{array} = \begin{array}{c} c : \text{---} \bullet \text{---} \bullet \text{---} \\ t : \text{---} [\Pi_1] [\Pi_2] \text{---} \end{array} \quad (2)$$

In order to add another control, we can use Lemma 2 to apply an axis transformation and implement Π_2 as $\Pi_3 \Pi_1 \Pi_3$, defined by $\hat{v}_3$ on the same plane as $\hat{v}_1, \hat{v}_2$, and satisfying $\angle_{\hat{v}_3}^{\hat{v}_1} = \angle_{\hat{v}_2}^{\hat{v}_3} = \frac{\lambda}{4}$. As can be seen in Circuit (3), qubit c_1 controls the axis transformation, such that for the $|0\rangle_{c_1}$ state, the controlled- Π_1 gates cancel out since they are Hermitian, and for the $|1\rangle_{c_1}$ state, Circuit (2) is applied on c_2, t .

$$\begin{array}{c} c_1 : \text{---} \bullet \text{---} \\ c_2 : \text{---} \bullet \text{---} \\ t : \text{---} [W] \text{---} \end{array} = \begin{array}{c} c_1 : \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ c_2 : \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ t : \text{---} [\Pi_1] [\Pi_3] [\Pi_1] [\Pi_3] \text{---} \end{array} \quad (3)$$

By adding a controlled phase gate on c_1, c_2 this structure can implement *any* controlled-controlled $U(2)$

(C^2U) gate using six CNOTs, i.e. at the same cost as a Toffoli. Now, we wish to implement the C^2U gate in linear nearest-neighbor (LNN) connectivity. The control condition can be rewritten as $\{(c_1 \oplus c_2), c_2\} = \{0, 1\}$ instead of $\{c_1, c_2\} = \{1, 1\}$. We would like to use c_2 and $c_1 \oplus c_2$ to control our Π gates, in order to remove the need to swap c_1 towards the target. We introduce the gate $[\Pi]_t^{c_1 \oplus c_2} = [X]_{c_2}^{c_1} [\Pi]_t^{c_2} [X]_{c_2}^{c_1}$ for this task. This time, we use the axis transformation to replace Π_2 with $\Pi_3 \Pi_2 \Pi_3$. In this case, Π_3 is controlled by $c_1 \oplus c_2$ and will change Π_2 to Π_1 only if $c_1 \oplus c_2 = 1$. Therefore, $[\Pi_3]_t^{c_1 \oplus c_2} [\Pi_2]_t^{c_2} [\Pi_3]_t^{c_1 \oplus c_2} [\Pi_1]_t^{c_2}$ implements the C^2W gate. This can be used to implement any C^2U gate by adding a controlled phase gate as $[\Pi_4]_{c_2}^{c_1} [P]_{c_1} [X]_{c_2}^{c_1}$, with some $\hat{v}_4 \perp \hat{z}$ and P a single-qubit phase gate. Noting that two CNOT gates cancel out, we reach the 8 CNOT implementation in Circuit (4), again providing the same CNOT count as a Toffoli [12, 26, 27, 28, 29], while generalizing the controlled operator to *any* $U(2)$ for a specific target location.

$$\begin{array}{c} c_1 : \text{---} \bullet \text{---} \\ c_2 : \text{---} \bullet \text{---} \\ t : \text{---} \boxed{U} \text{---} \end{array} = \begin{array}{c} c_1 : \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \boxed{P} \text{---} \\ c_2 : \text{---} \bullet \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \boxed{\Pi_4} \text{---} \\ t : \text{---} \boxed{\Pi_1} \text{---} \boxed{\Pi_3} \text{---} \boxed{\Pi_2} \text{---} \boxed{\Pi_3} \text{---} \end{array} \quad (4)$$

In case Circuit (4) is used in order to implement an $R_x(\lambda)$ gate with two controls, we can always choose $\Pi_3 = Z$, and define $\hat{v}_1, \hat{v}_2$ accordingly on the zy plane as $\hat{v}(\frac{\lambda}{4}, \frac{\pi}{2})$ and $\hat{v}(\frac{\lambda}{4}, -\frac{\pi}{2})$ respectively. Moreover, since $R_x \in SU(2)$, we get $\Pi_4 = X$, and the P gate can be removed. Using Lemma 4, we can decompose $[\Pi_1]_t^{c_2} = [\Pi_5]_t [Z]_t^{c_2} [\Pi_5]_t$, and $[\Pi_2]_t^{c_2} = [\Pi_6]_t [Z]_t^{c_2} [\Pi_6]_t$ with $\hat{v}_5 = \hat{v}(\frac{\lambda}{8}, \frac{\pi}{2})$ and $\hat{v}_6 = \hat{v}(\frac{\lambda}{8}, -\frac{\pi}{2})$. By adding a control to the axis transformation gates Π_5, Π_6 we achieve Circuit (5). We note that these controlled Π gates can be decomposed using one CZ and two R_x gates using Lemma 11.

$$\begin{array}{c} c_1 : \text{---} \bullet \text{---} \\ c_2 : \text{---} \bullet \text{---} \\ t : \text{---} \boxed{R_x} \text{---} \\ c_3 : \text{---} \bullet \text{---} \end{array} = \begin{array}{c} c_1 : \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ c_2 : \text{---} \bullet \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \\ t : \text{---} \boxed{\Pi_5} \text{---} \boxed{\Pi_5} \text{---} \boxed{\Pi_6} \text{---} \boxed{\Pi_6} \text{---} \\ c_3 : \text{---} \bullet \text{---} \end{array} \quad (5)$$

It can be noticed that the rightmost $[X]_{c_2}^{c_1} [Z]_t^{c_2} [X]_{c_2}^{c_1}$ is equivalent to $[Z]_t^{c_1} [Z]_t^{c_2}$ [30]. We can remove these gates when implementing a C^3R_x up-to a relative phase, which can therefore be implemented using 9 CNOT gates. In case the target is located at the bottom, the gate requires 11 CNOT gates as Circuit (6). Since any R_x can be decomposed as $Z\Pi_{\bar{x}}$, for $\Pi_{\bar{x}}$ defined according to Lemma 1 as a π -rotation about an axis on the zy plane, we note that Circuit (6) implements any chosen $C^3\Pi_{\bar{x}}$ up to a relative phase as well.

$$\begin{array}{c} \bullet \text{---} \boxed{\Delta'} \text{---} \\ \bullet \text{---} \boxed{R_x} \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \boxed{\Delta} \text{---} \\ \bullet \text{---} \boxed{\Pi_{\bar{x}}} \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \oplus \text{---} \\ \bullet \text{---} \boxed{\Pi_5} \text{---} \boxed{\Pi_5} \text{---} \boxed{\Pi_6} \text{---} \boxed{\Pi_6} \text{---} \end{array} \quad (6)$$

with Δ, Δ' as the relative phase (diagonal) gates. The boxed gates only require two CNOTs each as Circuit (7).

$$\begin{array}{c} \bullet \text{---} \boxed{\Pi} \text{---} \\ \bullet \text{---} \bullet \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \oplus \text{---} \boxed{\Pi'} \text{---} \\ \bullet \text{---} \oplus \text{---} \boxed{\Pi'} \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \oplus \text{---} \bullet \text{---} \boxed{\Pi'} \text{---} \\ \bullet \text{---} \oplus \text{---} \bullet \text{---} \end{array} \quad (7)$$

Noting that by choosing the target rotation as $R_x(\pi)$, or equivalently $\Pi_{\bar{x}} = Y$, Circuit (6) implements a 3-controlled Toffoli up-to a relative phase ($C^3X-\Delta$) at the same CNOT count reported by [12]. Our generalization of the $C^3X-\Delta$ gate to $C^3\Pi_{\bar{x}}-\Delta$ proves to be useful in the construction of the 4-controlled Toffoli (C^4X). First, we implement the C^4R_x gate as Circuit (8). It can be noted that the relative phase gates $\Delta, \Delta^\dagger$ can commute with the CZ gate between them and cancel out. Ignoring these gates, this circuit is similar to Circuit (3), with an axis transformation controlled by three qubits. Specifically, in this case Π_1 is replaced with Z , and Π_3 with $\Pi_{\bar{x}}$, thus allowing to implement any four-controlled R_x gate.

$$\begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \end{array} \quad (8)$$

Noting that this gate can be used for any chosen R_x rotation, we can choose $R_x(\pi) = -iX$. The $-i$ phase can be removed by applying a C^3S gate on the control qubits. We implement the C^4X gate as Circuit (9), applying SWAP gates to place the target of our C^4R_x gate at the bottom. Each SWAP gate only requires two CNOT gates instead of three, since a CNOT gate applied on the target of the C^4R_x gate commutes with it.

$$\begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \oplus \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \oplus \text{---} \end{array} = \begin{array}{c} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ \bullet \text{---} \oplus \text{---} \end{array} \quad (9)$$

Finally, noting that the leftmost CZ in Circuit (8), and one CNOT gate from Circuit (9) can be merged to one controlled π -rotation as $[Z]_t^c [X]_t^c = [iY]_t^c$, these can be implemented using one CNOT gate. We get that the C^4R_x gate with a target at the bottom requires 27 CNOT gates, as it uses two $C^3\Pi_{\bar{x}}-\Delta$ gates and five $C\Pi$ gates. Since the C^3S gate can be implemented using 18 CNOT gates as a C^3V [12] with two Hadamard gates, we managed to reduce the cost of the C^4X (aka the 5q Toffoli) gate in LNN connectivity from 48 [12, 31] to 45. As the best known decomposition of the C^4X gate without ancilla in unrestricted connectivity requires 30 CNOT gates [12, 32], we effectively reduce the CNOT overhead required to map this gate to

LNN connectivity by 16.6%. We show in [25, 18] that these methods result in reduced CNOT count for any number of controls, both for LNN and for unrestricted connectivity.

7 Controlled $U(4)$ gates

In this section we focus on the implementation of controlled two-qubit operators $[V]_{\{t_1, t_2\}}^c$ with any $V \in U(4)$. Our goal is to minimize the number of CNOT gates used. The approach we take demonstrates the ability to "add a control" to a given circuit (in this case an arbitrary two-qubit gate) by converting single qubit rotations to Hermitian gates, and adding a control to these gates at a reduced cost using Theorem 1. First we show that the following structure can implement any $V \in U(4)$ gate up to a global phase:

$$\begin{aligned} t_1 : & \text{---} \Pi_1^z \text{---} U_1 \text{---} \Pi_3^y \text{---} \Pi_5^y \text{---} U_1^\dagger \text{---} \Pi_6 \text{---} \\ t_2 : & \text{---} \Pi_2^z \text{---} U_2 \text{---} iY \text{---} \Pi_4^z \text{---} \oplus \text{---} U_2^\dagger \text{---} \Pi_7 \text{---} \end{aligned} \quad (10)$$

With $U_1, U_2 \in SU(2)$, and $\Pi(\hat{v}_{j \in \{1, 2, \dots, 7\}})$ gates are marked as Π_j^σ if it is known that $\hat{v}_j \perp \hat{\sigma}$ or as Π_j otherwise. We note that the Π_j^σ gates require one parameter, the Π_j require two parameters, and each $SU(2)$ gate and its inverse use the same three parameters, therefore this circuit uses 15 parameters which is the minimal requirement for a $U(4)$ operator disregarding the global phase.

Lemma 5. *Any $V \in U(4)$ gate can be implemented up to a global phase using Circuit (10), such that $\hat{v}_3, \hat{v}_5$ are on the xz plane, and $\hat{v}_1, \hat{v}_2, \hat{v}_4$ are on the xy plane.*

Proof. The following circuit is known to implement any $V \in U(4)$ up to a global phase:

$$\begin{aligned} t_1 : & \text{---} D \text{---} \bullet \text{---} R_{\hat{y}}(\theta_1) \text{---} \oplus \text{---} R_{\hat{y}}(\theta_2) \text{---} \bullet \text{---} B \text{---} \\ t_2 : & \text{---} C \text{---} \oplus \text{---} R_{\hat{z}}(\theta_3) \text{---} \bullet \text{---} \oplus \text{---} A \text{---} \end{aligned} \quad (11)$$

where the angles $\theta_1, \theta_2, \theta_3$, and the operators $A, B, C, D \in SU(2)$ can be found as shown in [33, 34, 35] for any chosen V . The middle CNOT gate can be replaced by $[H]_{t_1}[Z]_{t_1}^{\dagger}[H]_{t_1}$, and the first CNOT by $[S^\dagger]_{t_1}[\Pi_S]_{t_2}[iY]_{t_2}^{\dagger}[\Pi_S]_{t_2}$ from Lemma 4. The following circuit therefore implements V up to a global phase:

$$\begin{aligned} t_1 : & \text{---} D' \text{---} \bullet \text{---} R_{\hat{y}}(\theta_1) \text{---} H \text{---} \bullet \text{---} H \text{---} R_{\hat{y}}(\theta_2) \text{---} \bullet \text{---} B \text{---} \\ t_2 : & \text{---} C' \text{---} iY \text{---} \Pi_S \text{---} R_{\hat{z}}(\theta_3) \text{---} \bullet \text{---} \oplus \text{---} A \text{---} \end{aligned} \quad (12)$$

with $D' := S^\dagger D$, and $C' := \Pi_S C$. The gate $R_{\hat{z}}(\theta_3)$ can be decomposed as $\Pi(\hat{v}_4)\Pi_S$ with $\hat{v}_4 = \hat{v}(\frac{\pi}{2}, \frac{\pi}{4} + \frac{\theta_3}{2})$ using Lemma 1. Similarly, $HR_{\hat{y}}(\theta_1) = \Pi(\hat{v}_3)$, and $R_{\hat{y}}(\theta_2)H = \Pi(\hat{v}_5)$ with $\hat{v}_3 = \hat{R}_{\hat{y}}(-\frac{\theta_1}{2})\hat{h}$, $\hat{v}_5 = \hat{R}_{\hat{y}}(\frac{\theta_2}{2})\hat{h}$ such that $\Pi(\hat{h}) = H$ for $\hat{h} = \hat{v}(\frac{\pi}{4}, 0)$. We note that $\hat{v}_3, \hat{v}_5$ are on the xz plane, and $\hat{v}_4$ is on the xy plane.

Finally, we define $U_{BD} := BD'$. This operator can be expressed as $U_{BD} = R_{\hat{v}_B}(\theta_B)$ up to a global phase with a unit vector $\hat{v}_B$ and an angle θ_B . U_{BD} can therefore be decomposed using Lemma 1 as $\Pi(\hat{v}_6)\Pi(\hat{v}_1)$. The vector $\hat{v}_1$ can be chosen to be perpendicular to both $\hat{v}_B$ and $\hat{z}$ such that it is on the xy plane, and $\hat{v}_6 = \hat{R}_{\hat{v}_B}(\frac{\theta_B}{2})\hat{v}_1$. Since Π gates are Hermitian we can write $\Pi(\hat{v}_6)BD'\Pi(\hat{v}_1) = I$. Defining $U_1 = D'\Pi(\hat{v}_1)$ up to a global phase, such that $U_1 \in SU(2)$, therefore provides $U_1^\dagger = \Pi(\hat{v}_6)B$. Repeating this for A, C' produces $U_2, U_2^\dagger, \hat{v}_2, \hat{v}_7$ such that $\hat{v}_2$ is on the xy plane. $\square$

Circuit (13) is achieved by replacing each Π gate in Circuit (10) with its controlled version.

$$\begin{aligned} c : & \text{---} \bullet \text{---} P(\phi) \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \text{---} \\ t_1 : & \text{---} \Pi_1^z \text{---} U_1 \text{---} \bullet \text{---} \Pi_3^y \text{---} \Pi_5^y \text{---} U_1^\dagger \text{---} \Pi_6 \text{---} \\ t_2 : & \text{---} \Pi_2^z \text{---} U_2 \text{---} iY \text{---} \Pi_4^z \text{---} \oplus \text{---} U_2^\dagger \text{---} \Pi_7 \text{---} \end{aligned} \quad (13)$$

We show in Appendix A that this circuit, which requires ten controlled π -rotations, can implement any $[V]_{\{t_1, t_2\}}^c$ gate. Circuits for LNN restricted connectivity are provided as well. In addition, we present methods to efficiently convert a circuit which was designed using controlled π -rotation gates to other sets of universal gates. These methods are then used to decompose our Controlled $U(4)$ implementation to CNOT gates, along with arbitrary rotations, π rotations or $\{R_z, R_y\}$ gates. The results are summarized in Table 2.

Connectivity	CNOT	$R_{\hat{v}}/\Pi/R_{\hat{z}, \hat{y}}$
All-to-all	10	15/20/25
LNN	13	15/20/25

Table 2: Counts of CNOT gates and single-qubit gates from three universal gate sets required for our implementation of any Controlled- $U(4)$ gate in all-to-all or LNN connectivity.

8 Conclusion

In this paper, we explored the possibility and benefits of doing quantum computation purely by using single and two qubit Hermitian gates. Specifically, π -rotations about two fixed axes, along with the CNOT gate are universal for quantum computation. When arbitrary axes are available, any single qubit gate can be decomposed as two π -rotations. As shown here, this decomposition is particularly beneficial for single-qubit state preparation in the presence of systematic amplitude errors, while an alternative sequence of four π -rotations allows to improve the fidelity of arbitrary single-qubit gates.

In addition, we have shown that two Π gates can be used in order to transform the rotation axis of multi-controlled $U(2)$ gates. It follows that the CNOT count

of an n -controlled $SU(2)$ gate is independent of the choice of rotation axis. Therefore, the CNOT count of $20n$ reported by [23] immediately reduces to $16n$, which has been shown to be sufficient for specific chosen axes by the same paper. This axis transformation also allows to implement any singly-controlled Π gate using one CNOT gate.

The universality of the controlled Π gates, along with their reduced CNOT count, provides a convenient framework for circuit optimization and compilation. We have demonstrated some benefits of this approach while optimizing controlled $U(4)$ gates, resulting in a low number of CNOT gates for both all-to-all and LNN architectures. In the process, we have provided methods that allow one to efficiently convert a circuit which has been constructed using controlled Π gates to another set of gates. In addition, we used this framework to implement various multi-controlled single-target gates with a small number of controls, while either reproducing or generalizing state-of-the-art methods. For the case of the 5q Toffoli gate without ancilla in LNN connectivity, we managed to reduce the CNOT count from 48 [12] to 45. This reduction becomes especially advantageous in circuits which use many instances of such gates. Additional cases in which the use of Hermitian gates can provide efficient implementations of quantum operators have been covered in [25, 18].

Focusing on Hermitian gates may show benefits when implemented directly as π rotations using composite pulses, in addition to providing convenient tools for circuit optimization tasks.

For future research, one may consider the situation in which the strength of two qubit interactions can be so adjusted that the entangling part/s of a CNOT require the same time as a π rotation on a single qubit. Under these circumstances, as local gates can always be accomplished as π rotations, every operation can be accomplished by repeating the same pulse. Thus, π pulses applied by a control machine may be applied as a square wave with constant frequency, similarly to a clock used for classical computation. This can potentially make it easier to establish a clocked version of quantum computation.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/R513143/1, EP/T517793/1].

References

- [1] Michael A. Nielsen and Isaac Chuang. “Quantum Computation and Quantum Information”. *American Journal of Physics* **70**, 558–559 (2002).
- [2] N. Bar-Gill, L. M. Pham, A. Jarmola, D. Budker, and R. L. Walsworth. “Solid-state electronic spin coherence time approaching one second”. *Nature Communications* **4**, 1743 (2013).
- [3] Serge Haroche. “Collège de France abroad Lectures Quantum information with real or artificial atoms and photons in cavities” (2012). Centre for Quantum Technologies, National University of Singapore. <https://www.cqt.sg/highlight/2012-03-college-de-france-lectures-serge-haroche/>.
- [4] Sami Husain, Minaru Kawamura, and Jonathan A. Jones. “Further analysis of some symmetric and antisymmetric composite pulses for tackling pulse strength errors”. *Journal of Magnetic Resonance* **230**, 145–154 (2013).
- [5] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. “Optimal arbitrarily accurate composite pulse sequences”. *Physical Review A* **89**, 022341 (2014).
- [6] R. Tycko, A. Pines, and J. Guckenheimer. “Fixed point theory of iterative excitation schemes in NMR”. *The Journal of Chemical Physics* **83**, 2775–2802 (1985).
- [7] Jonathan A. Jones. “Nested composite NOT gates for quantum computation”. *Physics Letters A* **377**, 2860–2862 (2013).
- [8] Lorenza Viola, Emanuel Knill, and Seth Lloyd. “Dynamical Decoupling of Open Quantum Systems”. *Physical Review Letters* **82**, 2417–2421 (1999).
- [9] Götz S. Uhrig. “Keeping a Quantum Bit Alive by Optimized π -Pulse Sequences”. *Physical Review Letters* **98**, 100504 (2007).
- [10] Alexandre M. Souza, Gonzalo A. Álvarez, and Dieter Suter. “Robust Dynamical Decoupling for Quantum Computing and Quantum Memory”. *Physical Review Letters* **106**, 240501 (2011).
- [11] Alexandre M. Souza, Gonzalo A. Álvarez, and Dieter Suter. “Robust dynamical decoupling”. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **370**, 4748–4769 (2012).
- [12] Nikita A. Nemkov, Evgeniy O. Kiktenko, Ilia A. Luchnikov, and Aleksey K. Fedorov. “Efficient variational synthesis of quantum circuits with coherent multi-start optimization”. *Quantum* **7**, 993 (2023).
- [13] Alston S. Householder. “Unitary Triangularization of a Nonsymmetric Matrix”. *Journal of the ACM* **5**, 339–342 (1958).
- [14] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and

- Harald Weinfurter. “Elementary gates for quantum computation”. *Physical Review A* **52**, 3457–3467 (1995).
- [15] Danail Brezov, Clementina Mladenova, and Ivailo Mladenov. “Vector Decompositions of Rotations”. *Journal of Geometry and Symmetry in Physics* **28**, 67–103 (2012).
- [16] V. D. Donchev, C. D. Mladenova, and I. M. Mladenov. “On the compositions of rotations”. *AIP Conference Proceedings* **1684**, 080004 (2015).
- [17] Sergey Bravyi and Alexei Kitaev. “Universal quantum computation with ideal Clifford gates and noisy ancillas”. *Physical Review A* **71**, 022316 (2005).
- [18] Ben Zindorf and Sougato Bose. “Multi-Controlled Quantum Gates in Linear Nearest Neighbor” (2025). [arXiv:2506.00695](#).
- [19] Stephen Wimperis. “Iterative schemes for phase-distortionless composite 180° pulses”. *Journal of Magnetic Resonance* (1969) **93**, 199–206 (1991).
- [20] S. Wimperis. “Broadband, Narrowband, and Passband Composite Pulses for Use in Advanced NMR Experiments”. *Journal of Magnetic Resonance, Series A* **109**, 221–231 (1994).
- [21] Hayk L. Gevorgyan and Nikolay V. Vitanov. “Ultrahigh-fidelity composite rotational quantum gates”. *Physical Review A* **104**, 012609 (2021).
- [22] Yun-Pil Shim, Jianjia Fei, Sangchul Oh, Xuedong Hu, and Mark Friesen. “Single-qubit gates in two steps with rotation axes in a single plane” (2013). [arXiv:1303.0297](#).
- [23] Rafaella Vale, Thiago Melo D. Azevedo, Ismael C. S. Araújo, Israel F. Araujo, and Adenilton J. da Silva. “Circuit Decomposition of Multi-controlled Special Unitary Single-Qubit Gates”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **43**, 802–811 (2024).
- [24] Evandro C. R. Rosa, Eduardo I. Duzzioni, and Rafael de Santiago. “Optimizing Gate Decomposition for High-Level Quantum Programming”. *Quantum* **9**, 1659 (2025).
- [25] Ben Zindorf and Sougato Bose. “Efficient implementation of multicontrolled quantum gates”. *Physical Review Applied* **24**, 044030 (2025).
- [26] Jan Gwinner, Marcin Briński, Wojciech Burkot, Lukasz Czerwiński, and Vladyslav Hlembotskyi. “Benchmarking 16-element quantum search algorithms on superconducting quantum processors” (2021). [arXiv:2007.06539](#).
- [27] Ken M. Nakanishi, Takahiko Satoh, and Synge Todo. “Quantum-gate decomposer” (2021). [arXiv:2109.13223](#).
- [28] Ken M. Nakanishi, Takahiko Satoh, and Synge Todo. “Decompositions of multiple controlled- Z gates on various qubit-coupling graphs”. *Physical Review A* **110**, 012604 (2024).
- [29] Pedro M. Q. Cruz and Bruno Murta. “Shallow unitary decompositions of quantum Fredkin and Toffoli gates for connectivity-aware equivalent circuit averaging”. *APL Quantum* **1**, 016105 (2024).
- [30] Dmitri Maslov and Ben Zindorf. “Depth Optimization of CZ, CNOT, and Clifford Circuits”. *IEEE Transactions on Quantum Engineering* **3**, 1–8 (2022).
- [31] Carolina Allende, André Fonseca de Olivera, and Efrain Buksman. “Synthesis of quantum circuits based on supervised learning and correlations”. *Quantum Information Processing* **23**, 204 (2024).
- [32] V.V. Shende, S.S. Bullock, and I.L. Markov. “Synthesis of quantum-logic circuits”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25**, 1000–1010 (2006).
- [33] Vivek V. Shende, Igor L. Markov, and Stephen S. Bullock. “Minimal universal two-qubit controlled-NOT-based circuits”. *Physical Review A* **69**, 062321 (2004).
- [34] Farrokh Vatan and Colin Williams. “Optimal Quantum Circuits for General Two-Qubit Gates”. *Physical Review A* **69**, 032315 (2004).
- [35] G. Vidal and C. M. Dawson. “Universal quantum circuit for two-qubit transformations with three controlled-NOT gates”. *Physical Review A* **69**, 010301 (2004).

Appendix

A Controlled $U(4)$ gates construction

We provide a constructive proof for our controlled- $U(4)$ structure which is achieved by 'adding a control' to Circuit (10).

Lemma 6. *Any operator which can be implemented exactly using Circuit (10) is in $SU(4)$.*

Proof. Any circuit over two qubits which is composed solely of $SU(2)$ and $SU(4)$ operator implements an $SU(4)$ operator. Therefore, we simply express each gate in Circuit (10) as a $SU(2)$ or $SU(4)$ operator multiplied by a phase, and show that the total added phase equals to 1. By definition, $\Pi(\hat{v}) = iR_{\hat{v}}(\pi)$ such that $R_{\hat{v}}(\pi) \in SU(2)$. The $[iY]_{t_2}^{t_1}$ gate is in $SU(4)$ since its determinant equals 1. The CNOT and CZ gates have a determinant of -1 , and therefore can be written as $[X]_{t_2}^{t_1} = \sqrt{i}G$ and $[Z]_{t_2}^{t_1} = \sqrt{i}F$ with $G, F \in SU(4)$. The rest of the gates are $SU(2)$ operators, and therefore, the total added phase is $i^7 \sqrt{i} \sqrt{i} = 1$. $\square$

Lemma 7. *Any $[V]_{\{t_1, t_2\}}^c$ gate can be implemented with 10 CNOT gates using Circuit (13), s.t. $\hat{v}_3, \hat{v}_5$ are on the xz plane, $\hat{v}_1, \hat{v}_2, \hat{v}_4$ are on the xy plane, and $V \in U(4)$.*

Proof. According to Lemma 5 and Lemma 6, for any operator $V \in U(4)$, there exists an operator $U \in SU(4)$ which can be implemented exactly using Circuit (10) such that $V = e^{i\phi}U$. Since the determinant of any $SU(4)$ operator equals 1, the angle $\phi \in [0, 2\pi)$ must satisfy $e^{i4\phi} = \det(V)$. Thus, there are four options for the value of ϕ , and it will be chosen as one which allows U to be implemented exactly using Circuit (10). The parameters defining $U_1, U_2 \in SU(2)$, and $\hat{v}_{j \in [1, 7]}$ can be found as shown in Lemma 5 such that $\hat{v}_3, \hat{v}_5$ are on the xz plane, $\hat{v}_1, \hat{v}_2, \hat{v}_4$ are on the xy plane. We investigate the operator applied on t_1, t_2 by Circuit (13) for each computational basis state of the control qubit c .

For the state $|1\rangle$ of the control qubit c , the operator applied on qubits t_1, t_2 is described by Circuit (10), multiplied by a phase $e^{i\phi}$, and is therefore $e^{i\phi}U = V$.

For the state $|0\rangle$ of the control qubit c , the operator applied on qubits t_1, t_2 by Circuit (13) is as follows:

$$\begin{array}{c} t_1: \text{---} [U_1] \text{---} \bullet \text{---} \bullet \text{---} [U_1^\dagger] \text{---} \\ t_2: \text{---} [U_2] \text{---} [iY] \text{---} \oplus \text{---} [U_2^\dagger] \text{---} \end{array} = \begin{array}{c} t_1: \text{---} [U_1] \text{---} \bullet \text{---} [U_1^\dagger] \text{---} \\ t_2: \text{---} [U_2] \text{---} [iXZY] \text{---} [U_2^\dagger] \text{---} \end{array}$$

Since $iXZY = I$, and $U_1^\dagger U_1 = U_2^\dagger U_2 = I$, this circuit implements the 4×4 identity matrix.

Therefore, Circuit (13) can implement any controlled $V \in U(4)$ operator. The circuit requires ten $[e^{i\psi}\Pi]_t^c$ gates, each requires one CNOT according to Theorem 1. $\square$

We now provide implementations of the controlled $U(4)$ gate for LNN connectivity.

Lemma 8. *Any $[V]_{\{t_1, t_2\}}^c$ gate with $V \in U(4)$ can be implemented in LNN architecture over 3 neighbouring qubits using 13 CNOT gates if c is the first or the last qubit.*

Proof. According to Lemma 7, any $[V]_{\{t_1, t_2\}}^c$ operator can be implemented using the structure described in Circuit (13) if c is the first qubit (if it is the last qubit then the structure is simply flipped). This structure requires three gates of the form $[\Pi_b]_{t_2}^c [\Pi_a]_{t_1}^c$. Each of these can be decomposed in LNN connectivity using 3 CNOT gates simply by decomposing the gates as Circuit (1) and applying the identity $[X]_{t_1}^c [X]_{t_2}^c = [X]_{t_2}^{t_1} [X]_{t_1}^c [X]_{t_2}^{t_1}$. The circuit requires 4 additional CNOT gates which already operate on neighbouring qubits. $\square$

Lemma 9. *Any $[V]_{\{t_1, t_2\}}^c$ gate with $V \in U(4)$ can be implemented in LNN architecture over 3 neighbouring qubits using 13 CNOT gates if c is the central qubit.*

Proof. The labels of c, t_1 in Circuit (13) can be swapped by applying a SWAP gate on each side of the circuit.

The gates $[\Pi_1]_c^{t_1} [\Pi_2]_{t_2}^{t_1} [\text{SWAP}]_c^{t_1}$ can be applied using 3 CNOT gates using $[\Pi_2]_{t_2}^{t_1} [\text{SWAP}]_c^{t_1} = [\text{SWAP}]_c^{t_1} [\Pi_2]_{t_2}^c$, and $[\Pi_1]_c^{t_1} [\text{SWAP}]_c^{t_1} = [\Pi_1']_c [X]_{t_1}^{t_1} [X]_c^{t_1} [\Pi_1']_{t_1}$, with Π_1' obtained using the decomposition of $[\Pi_1]_c^{t_1}$ as Circuit (1). The gates $[\text{SWAP}]_c^{t_1} [\Pi_7]_{t_2}^{t_1} [\Pi_6]_c^{t_1}$ are applied similarly. The remaining controlled gates are decomposed as in Lemma 8. $\square$

While π -rotations can be useful when implemented directly, some physical qubit implementations do not support such rotations, and in that case $\Pi(\hat{v})$ gates must be implemented using the available set of basic gates. However, as demonstrated in this paper, the π -rotation formalism can be useful as a convenient framework for circuit design. In this section we provide a few methods which can be used in many cases in order to convert a circuit which was built using π -rotations to an equivalent circuit which uses gates from the group $\{CNOT, R_{\hat{y}}, R_{\hat{z}}\}$, and use those methods in order to decompose the presented controlled $U(4)$ gates.

Lemma 10. *Any $[\Pi(\hat{v})]_t^c$ gate can be implemented as $[R_{\hat{z}}(\phi)]_t [R_{\hat{y}}(\psi)]_t [X]_t^c [R_{\hat{y}}(-\psi)]_t [R_{\hat{z}}(-\phi)]_t$ for some angles ϕ, ψ .*

Proof. According to Lemma 4, $[\Pi(\hat{v})]_t^c = [\Pi(\hat{v}_M)]_t [X]_t^c [\Pi(\hat{v}_M)]_t$ with $\hat{v}_M \in M(\hat{v}, \hat{x})$. Any operator can be decomposed using ZYX Euler (Tait–Bryan) angles, and therefore, three angles ϕ, ψ, ζ can be found such that $\Pi(\hat{v}_M) = R_{\hat{z}}(\phi) R_{\hat{y}}(\psi) R_{\hat{x}}(\zeta)$, and since $\Pi(\hat{v}_M)$ is Hermitian, $\Pi(\hat{v}_M) = R_{\hat{x}}(-\zeta) R_{\hat{y}}(-\psi) R_{\hat{z}}(-\phi)$ as well. The rotations around $\hat{x}$ can commute with the $[X]_t^c$ gate and cancel out, providing the required expression. $\square$

The number of single qubit rotations can be reduced in the specific cases in which $\hat{v}$ is perpendicular to the $\hat{x}$, $\hat{y}$ or $\hat{z}$ axis.

Lemma 11. *If $\hat{v} = \hat{R}_{\hat{\sigma}}(\phi)\hat{\tau}$ for any unit vectors $\hat{\tau}, \hat{\sigma}$ satisfying $\hat{\tau} \perp \hat{\sigma}$, then $[\Pi(\hat{v})]_t^C = [R_{\hat{\sigma}}(\phi)]_t[\Pi(\hat{\tau})]_t^C[R_{\hat{\sigma}}(-\phi)]_t$.*

Proof. Using Lemma 4, $[\Pi(\hat{v})]_t^C = [\Pi(\hat{v}_M)]_t[\Pi(\hat{\tau})]_t^C[\Pi(\hat{v}_M)]_t$ with $\hat{v}_M \in M(\hat{v}, \hat{\tau})$. Since both $\hat{v}, \hat{\tau}$ are perpendicular to $\hat{\sigma}$ and $\hat{v} = \hat{R}_{\hat{\sigma}}(\phi)\hat{\tau}$, we can choose $\hat{v}_M = \hat{R}_{\hat{\sigma}}(\frac{\phi}{2})\hat{\tau}$. This vector is on the same plane as $\hat{v}, \hat{\tau}$, and $\angle_{\hat{v}_M}^{\hat{v}} = \angle_{\hat{v}_M}^{\hat{\tau}}$ and therefore it satisfies $\hat{v} = \hat{R}_{\hat{v}_M}(\pi)\hat{\tau}$. Substituting $[\Pi(\hat{\tau})]_t^C = [\Pi(\hat{\tau})]_t[\Pi(\hat{\tau})]_t^C[\Pi(\hat{\tau})]_t$ provides:

$$[\Pi(\hat{v})]_t^C = [\Pi(\hat{v}_M)]_t[\Pi(\hat{\tau})]_t[\Pi(\hat{\tau})]_t^C[\Pi(\hat{\tau})]_t[\Pi(\hat{v}_M)]_t$$

From Lemma 1, $\Pi(\hat{v}_M)\Pi(\hat{\tau}) = R_{\hat{\sigma}}(\phi)$ and $\Pi(\hat{\tau})\Pi(\hat{v}_M) = R_{\hat{\sigma}}(-\phi)$. $\square$

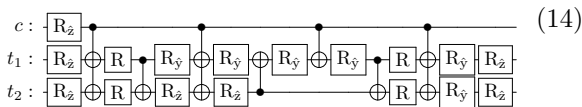
Using Lemma 11 with $\hat{\tau} = \hat{x}$ and $\hat{\sigma} \in \{\hat{y}, \hat{z}\}$ allows one to implement any single-controlled π -rotation about a unit vector on the xy or xz plane using two $R_{\hat{z}}$ or two $R_{\hat{y}}$ gates, respectively, along with one CNOT gate.

We now wish to decompose Circuit (13) using gates from the universal sets $\{CNOT, R_{\hat{y}}, R_{\hat{z}}\}$, $\{CNOT, R_{\hat{v}}\}$ and $\{CNOT, \Pi\}$. For each of these sets, we construct a circuit which can implement any controlled- $U(4)$ operator, and present the cost of the circuit in all-to-all and in LNN architectures.

Theorem 2. *Any $[V]_{\{t_1, t_2\}}^c, V \in U(4)$ gate can be implemented over three neighbouring qubits using 10 $R_{\hat{y}}$ gates and 15 $R_{\hat{z}}$ gates, in addition to 10 or 13 CNOT gates in all-to-all or LNN connectivity, respectively.*

Proof. According to Lemma 7, any $[V]_{\{t_1, t_2\}}^c, V \in U(4)$ gate can be implemented using Circuit (13) such that $\hat{v}_3, \hat{v}_5$ are on the xz plane, and $\hat{v}_1, \hat{v}_2, \hat{v}_4$ are on the xy plane. We can simply decompose Circuit (13) in terms of $\{CNOT, R_{\hat{y}}, R_{\hat{z}}\}$ gates.

First we note that $\hat{y} = \hat{R}_{\hat{z}}(\frac{\pi}{2})\hat{x}$, and $\hat{z} = \hat{R}_{\hat{y}}(-\frac{\pi}{2})\hat{x}$, thus according to Lemma 11, $[iY]_{t_2}^{t_1} = [S]_{t_2}[X]_{t_2}^{t_1}[S^\dagger]_{t_2}[S]_{t_1}$, and $[Z]_{t_1}^{t_2} = [R_{\hat{y}}(-\frac{\pi}{2})]_{t_2}[X]_{t_1}^{t_2}[R_{\hat{y}}(\frac{\pi}{2})]_{t_2}$. Any other controlled gate in the circuit can be implemented according to Lemma 11 if the axis is constrained to a specific plane, or according to Lemma 10 otherwise. Applying these to all gates and merging sequential rotations into one rotation provides the following structure:



where we only indicate the axis of rotation if known, and do not specify the angles for simplicity. This circuit requires four arbitrary rotations which cost 4 $R_{\hat{y}}$

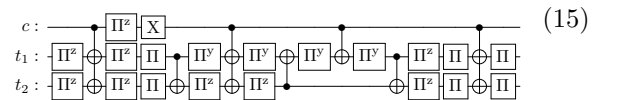
gates and 8 $R_{\hat{z}}$ gates using the ZYZ decomposition. In addition this circuit uses 6 $R_{\hat{y}}$ gates, 7 $R_{\hat{z}}$ gates and 10 CNOTs.

Following the steps in Lemma 8 and Lemma 9, it is clear that single qubit rotations are not added in the process of converting this circuit into its LNN versions which require 13 CNOT gates for any permutation over three neighbouring qubits. $\square$

The decomposition in terms of $\{CNOT, R_{\hat{v}}\}$ gates follows simply from Circuit (14), merging any consecutive rotations into one, and following the same logic of Theorem 2 for the LNN structures. Those steps show the correctness of Theorem 3.

Theorem 3. *Any $[V]_{\{t_1, t_2\}}^c, V \in U(4)$ gate can be implemented over three neighbouring qubits using 15 $R_{\hat{v}}$ gates, in addition to 10 or 13 CNOT gates in all-to-all or LNN connectivity respectively.*

Alternatively, Circuit (13) can be easily converted to a circuit which uses $\{CNOT, \Pi\}$ gates by repeating the same steps as the ones used to produce Circuit (14), but decomposing the last two controlled π -rotations according to Lemma 4. Then adding an X gate on both sides of each target of the first five CNOT gates which are controlled by qubit c allows to convert the $R_{\hat{z}}/R_{\hat{y}}$ gates to π -rotations with axis on the xy/xz plane using the same logic as in the proof of Lemma 11. The R gates are then decomposed according to Lemma 1, with one axis chosen from the xy plane, and the $R_{\hat{z}}$ gate on the control is decomposed similarly as $X\Pi^z$. The following circuit presents the decomposition in terms of the Hermitian universal set $\{CNOT, \Pi\}$.



This circuit uses 20 Π gates in total, while one is X and 13 are restricted to the xy or the xz plane. In the above circuit, it is implicit that according to the specific V , the axes corresponding to each of the Π gates will be determined. This circuit can be converted into its LNN versions in a similar manner to the proof of Theorem 2. Those steps show the correctness of Theorem 4.

Theorem 4. *Any $[V]_{\{t_1, t_2\}}^c, V \in U(4)$ gate can be implemented over three neighboring qubits using 20 Π gates, in addition to 10 or 13 CNOT gates in all-to-all or LNN connectivity, respectively.*

B Proofs of Π gate Lemmas

Here we provide proofs for Lemmas mentioned in the main text.

Lemma 1. Any $R_{\hat{v}}(\lambda) \in SU(2)$ operator can be implemented as $\Pi(\hat{v}_2)\Pi(\hat{v}_1)$ with $\hat{v}_1$ as any unit vector perpendicular to $\hat{v}$, and $\hat{v}_2 = \hat{R}_{\hat{v}}(\frac{\lambda}{2})\hat{v}_1$ with $\frac{\lambda}{2} \in (-\pi, \pi]$.

Proof. First, we focus on the case $\frac{\lambda}{2} \in (0, \pi)$ in which the ordered set $(\hat{v}_1, \hat{v}_2, \hat{v})$ is positively-oriented. Since $\hat{v}_1 \perp \hat{v}$, any rotation of $\hat{v}_1$ around $\hat{v}$ is perpendicular to $\hat{v}$ as well, and so $\hat{v}_2 \perp \hat{v}$, and $\angle_{\hat{v}_2}^{\hat{v}_1} = \frac{\lambda}{2}$. Therefore, using vector arithmetic, $\hat{v}_1 \cdot \hat{v}_2 = \cos \frac{\lambda}{2}$, and $\hat{v}_1 \times \hat{v}_2 = \hat{v} \sin \frac{\lambda}{2}$. The following is achieved by matrix multiplication, followed by substitutions provided by the above equations, e.g. $[v_{1x} v_{2y} - v_{1y} v_{2x}] = [v_z \sin \frac{\lambda}{2}]$.

$$\Pi(\hat{v}_2)\Pi(\hat{v}_1) = \begin{pmatrix} \cos \frac{\lambda}{2} - iv_z \sin \frac{\lambda}{2} & (-v_y - iv_x) \sin \frac{\lambda}{2} \\ (v_y - iv_x) \sin \frac{\lambda}{2} & \cos \frac{\lambda}{2} + iv_z \sin \frac{\lambda}{2} \end{pmatrix}$$

which is equivalent to $R_{\hat{v}}(\lambda)$ as shown in [1]. For $\frac{\lambda}{2} \in (-\pi, 0)$, the ordered set $(\hat{v}_1, \hat{v}_2, -\hat{v})$ is positively-oriented, and $\angle_{\hat{v}_2}^{\hat{v}_1} = |\frac{\lambda}{2}| = -\frac{\lambda}{2}$. Thus, using the above, $\Pi(\hat{v}_2)\Pi(\hat{v}_1) = \hat{R}_{-\hat{v}}(-\lambda) = R_{\hat{v}}(\lambda)$. For the specific cases of $\frac{\lambda}{2} \in \{0, \pi\}$, we get $R_{\hat{v}}(0) = \Pi(\hat{v}_1)\Pi(\hat{v}_1) = I$, and $R_{\hat{v}}(2\pi) = \Pi(-\hat{v}_1)\Pi(\hat{v}_1) = -I$ as expected. Therefore, the statement holds for $\frac{\lambda}{2} \in (-\pi, \pi]$. $\square$

Lemma 3. $R_{\hat{v}}(\lambda) = \Pi(\hat{v}_M)R_{\hat{v}'}(\lambda)\Pi(\hat{v}_M)$ for any angle λ , unit vectors $\hat{v}, \hat{v}'$, and $\hat{v}_M \in M(\hat{v}, \hat{v}')$.

Proof. According to Lemma 1, $R_{\hat{v}}(\lambda) = \Pi(\hat{v}_2)\Pi(\hat{v}_1)$, such that $\hat{v}_1 \perp \hat{v}$, and $\hat{v}_2 = \hat{R}_{\hat{v}}(\frac{\lambda}{2})\hat{v}_1$. Using Lemma 2, $\Pi(\hat{v}_j) = \Pi(\hat{v}_M)\Pi(\hat{v}_j')\Pi(\hat{v}_M)$ with $j \in \{1, 2\}$, and $\hat{v}_j' := \hat{R}_{\hat{v}_M}(\pi)\hat{v}_j$. Substituting and canceling out two $\Pi(\hat{v}_M)$ operators provides $R_{\hat{v}}(\lambda) = \Pi(\hat{v}_M)\Pi(\hat{v}_2')\Pi(\hat{v}_1')\Pi(\hat{v}_M)$.

Since the angles between vectors are conserved under rotations, and the same rotation matrix $\hat{R}_{\hat{v}_M}(\pi)$ is applied on $\hat{v}_1, \hat{v}_2, \hat{v}$ to produce $\hat{v}_1', \hat{v}_2', \hat{v}'$ respectively, we get that $\hat{v}_1' \perp \hat{v}'$, and $\hat{v}_2' = \hat{R}_{\hat{v}'}(\frac{\lambda}{2})\hat{v}_1'$. Therefore, $R_{\hat{v}'}(\lambda) = \Pi(\hat{v}_2')\Pi(\hat{v}_1')$. $\square$

Lemma 12. Any $[U]_t^c, U \in U(2)$ gate which can be implemented using one CNOT must satisfy $U = e^{i\psi}\Pi(\hat{v})$.

Proof. Any circuit over two qubits with one CNOT gate can be expressed in the following generic form:

$$[U]_t^c = [U_4]_t[U_3]_c[X]_t^c[U_1]_c[U_2]_t$$

The gates $U_{j \in \{1,2,3,4\}} \in U(2)$ can be written as $\begin{pmatrix} \alpha_j & \gamma_j \\ \beta_j & \delta_j \end{pmatrix}$. We adjust the equation and apply each side on $|00\rangle$.

$$[U_4]_t^c[U_1]_t^c[U_2]_t^c|00\rangle = [U_3]_c[X]_t^c[U_1]_c|00\rangle$$

Since $[U]_t^c$ applies the identity if the computational basis state of the control is $|0\rangle$, this is equivalent to

$$|0\rangle_c (U_4^\dagger U_2^\dagger |0\rangle)_t = [U_3]_c(\alpha_1 |00\rangle + \beta_1 |11\rangle) =$$

$$|0\rangle_c (\alpha_1 \alpha_3 |0\rangle + \beta_1 \gamma_3 |1\rangle)_t + |1\rangle_c (\alpha_1 \beta_3 |0\rangle + \beta_1 \delta_3 |1\rangle)_t$$

This provides the constraints $U_4^\dagger U_2^\dagger = \alpha_1 \alpha_3 I + \beta_1 \gamma_3 X$ and $|\alpha_1| |\beta_3| = |\beta_1| |\delta_3| = 0$. The identities $|\alpha_j| = |\delta_j|$, $|\beta_j| = |\gamma_j|$ and $|\alpha_j|^2 + |\beta_j|^2 = 1$ are used to show that there are only two possible solutions up to a global phase:

$$(U_4, U_3, U_1) = \begin{cases} (U_2^\dagger, P(\psi_3), P(\psi_1)) & \alpha_1 \neq 0 \\ (U_2^\dagger X, P(\psi_3')X, P(\psi_1')X) & \alpha_1 = 0 \end{cases}$$

And therefore any $[U]_t^c$ which can be implemented using a single CNOT can be written as

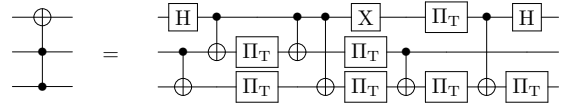
$$[U]_t^c = [P(\psi)]_c[U_2^\dagger]_t[X]_t^c[U_2]_t$$

this is obtained by substituting each of the solutions and using the identities $[X]_c[P(\psi)]_c = [P(-\psi)]_c[X]_c$, $[X]_t^c[P(\psi)]_c = [P(\psi)]_c[X]_t^c$ and $[X]_c[X]_t^c[X]_c[X]_t = [X]_t^c$.

Therefore, it must hold that $U = e^{i\psi}U_2^\dagger X U_2$. According to Lemma 1, we can find $\hat{v}_1, \hat{v}_2$ such that $U_2 = \Pi(\hat{v}_2)\Pi(\hat{v}_1)$, and so $U = e^{i\psi}\Pi(\hat{v}_1)\Pi(\hat{v}_2)X\Pi(\hat{v}_2)\Pi(\hat{v}_1)$. According to Lemma 2, one can find unit vectors $\hat{v}_2', \hat{v}$ such that $\Pi(\hat{v}_2)X\Pi(\hat{v}_2) = \Pi(\hat{v}_2')$, and $\Pi(\hat{v}_1)\Pi(\hat{v}_2')\Pi(\hat{v}_1) = \Pi(\hat{v})$. Therefore, $U = e^{i\psi}\Pi(\hat{v})$. $\square$

C Toffoli decompositions

The following circuit demonstrates the decomposition of the Toffoli gate using only gates from $\{CNOT, H, X, \Pi_T\}$.



This decomposition can be easily achieved by converting the well known implementation of the Toffoli gate, replacing T and $T^\dagger$ gates with $\Pi_T X$ and $X \Pi_T$ respectively, and canceling out X gates.

The following circuit implements the Toffoli gate up to a global phase using the minimal universal Hermitian set $\{CNOT, H, \Pi_T\}$ with seven CNOT, seven Π_T and two H gates.

