Clifford Approximation of Unitary Matrices

Kalle Volanto, Mahdi Bayanifar, Olav Tirkkonen
Aalto University, Helsinki, Finland
e-mails: {kalle.volanto, mahdi.bayanifar, olav.tirkkonen}@aalto.fi

***Abstract*—Approximating a given unitary operator into a discrete set of gates is of great importance in quantum computation. This problem can be seen as a source quantization problem where the unitary operator is considered as the source. Elements of the Clifford hierarchy can be implemented fault-tolerantly using quantum gate teleportation. This motivates the development of algorithms that quantize unitary operators into Clifford hierarchy elements. In this paper, we present a fast algorithm for approximating arbitrary n-qubit unitaries with Clifford matrices. The algorithm relies on the Clifford transvection decomposition of the Clifford group. The algorithm greedily applies Clifford transvections taking the input unitary closer to the identity matrix. To make the algorithm fast, we take advantage of the Pauli basis representation for unitary matrices to parallelize matrix distance computations. The complexity of the proposed algorithm is $O(n\,4^n)$. Based on simulation results, the proposed algorithm outperforms the state of the art method.**

***Index Terms*—Quantum computation, Source quantization, Clifford hierarchy, Transvection decomposition.**

I. INTRODUCTION

In fault tolerant universal quantum computation, it is an important task to find the closest approximation of a unitary quantum operator in a discrete gate set with a given circuit depth. For single-qubit unitary operators, the situation is well-understood [1], [2], [3], [4]. Universal computation can be approximated in a set consisting of single-qubit Clifford gates, together with the T-gate that realizes a $\pi/4$ rotation. For multi-qubit unitary operators, decomposition into a combination of two-qubit CNOT-gates and arbitrary, continuous-valued single-qubit gates is a nearly mature research problem [5], [6], [7]. The predominant method in the literature to approximate an arbitrary unitary is to first decompose into CNOT and arbitrary single-qubit gates, and then to approximate each of the single-qubit gates separately [8]. This approach represents the state of the art in approximating unitaries using a discrete gate set [9], where the block-ZXZ decomposition [7] is used as a component.

The Clifford hierarchy is of great importance for quantum computation and quantum error correction. Quantum gate teleportation [10], [11] offers a fault-tolerant way to implement operators from the Clifford hierarchy. The T-gate belongs to the third level of the Clifford hierarchy, which means that the resulting quantum circuit can be implemented fault-tolerantly. However, the implementation cost of a T-gate is much higher than the cost of Clifford-gates. In [12], it is argued that the overall T-count of a quantum circuit can be reduced if diagonal elements [13] in the Clifford hierarchy are directly considered.

From the perspective of coding theory and signal processing, the problem of approximating a unitary operator in a discrete gate set with a given circuit depth is a source coding problem. The unitary operator represents the source, which for universal quantum computing can be assumed to be uniformly (Haar) distributed across the space of unitary matrices. The set of gates created from a finite gate set and a given circuit depth gives rise to a discrete *codebook*, and the task of a source encoder is to find the element in the codebook that best approximates a given unitary. We thus have a source quantization problem, or equivalently a decoding problem, where the closest codeword in the codebook has to be identified from a distorted codeword. Accordingly, decoding algorithms from the literature on coding theory may be applied to the problem of approximating a unitary quantum operator.

This is particularly suitable when approximating gates from the Clifford hierarchy. Codebooks of vectors constructed from the Clifford group have been extensively studied [14], [15], [16], [17]. In [14], [15], codebooks of Binary Chirps were defined, with a cardinality growing as a power of its dimension $N = 2^n$, and a decoding algorithm with complexity $O(N(\log N)^2)$ is proposed. In [16], the Binary Chirps were found to be columns of a subset of the Clifford group that is isomorphic to the diagonal part of the group, and codebooks consisting of columns of generic Clifford group elements were considered. In [17], this approach was generalized to diagonal elements of the Clifford hierarchy.

Inspired by these works, a viable research direction is to develop low-complexity methods for directly identifying the element in the Clifford hierarchy which is closest to a unitary operator. In this paper, we start this program by considering the second level in the hierarchy, i.e., the Clifford group. The aim of this paper is to develop ideas which can be used for the problem with higher levels of the full Clifford hierarchy. For more than 2 qubits, the Clifford group size is too large for the exhaustive search to be reasonable [18], so efficient approximation algorithms are needed.

We use as a starting point the fact that the Clifford group can be generated using Clifford transvections, and draw inspiration from the corresponding transvection decomposition of the Clifford group elements [19]. We extend this to approximate arbitrary unitaries. The idea is that multiplying an arbitrary unitary with specific transvections results in a matrix which is close to the identity matrix, giving a good approximation from the Clifford group. To make the algorithm fast, we utilize the Pauli basis representation for unitary matrices. Inspired by [14], we compute several distances in parallel using the Hadamard transform, which reduces the complexity order of the algorithm from $O(N^3)$ to $O(N^2 \log N)$. By simulation, we show that when approximating multi-qubit unitary operators, the developed algorithm outperforms a state of the art method based on the block-ZXZ Decomposition with a considerable margin.

II. PRELIMINARIES

A. Unitaries and Projective Unitaries

An n-qubit operator can be represented as the matrix $\mathbf{U} \in \mathbb{U}(N)$, where $N = 2^n$, and $\mathbb{U}(N)$ is the group of unitary $N \times N$ matrices [20]. The global phase of the quantum state is not observable. Therefore the unitary operators in this paper are thought of as elements in the projective unitary group

$$\mathrm{P}\mathbb{U}(N) = \mathbb{U}(N)/\mathbb{U}(1). \tag{1}$$

B. Pauli Group

The single-qubit Pauli group is generated by the single-qubit gates [20]

$$\mathbf{X} := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } \mathbf{Z} := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2}$$

For binary n-dimensional vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n$, define $\mathbf{D}(\mathbf{a}, \mathbf{b})$ as the Kronecker product

$$\mathbf{D}(\mathbf{a}, \mathbf{b}) := \mathbf{X}^{a_1}\mathbf{Z}^{b_1} \otimes \cdots \otimes \mathbf{X}^{a_n}\mathbf{Z}^{b_n}, \tag{3}$$

where a_i and b_i for $i = 1, 2..., n$ denotes ith element of $\mathbf{a}$ and $\mathbf{b}$, respectively. The Pauli group for n qubits is of the form

$$\mathcal{P}_n := \{ i^k \mathbf{D}(\mathbf{a}, \mathbf{b}) \mid \mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n, k \in \{0, 1, 2, 3\}\}. \tag{4}$$

Given $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{F}_2^n$, we have

$$\mathbf{D}(\mathbf{a}, \mathbf{b})\mathbf{D}(\mathbf{c}, \mathbf{d}) = (-1)^{\mathbf{b}^T\mathbf{c}}\mathbf{D}(\mathbf{a} + \mathbf{c}, \mathbf{b} + \mathbf{d}). \tag{5}$$

The Hermitian transpose of $\mathbf{D}(\mathbf{a}, \mathbf{b})$ is

$$\mathbf{D}(\mathbf{a}, \mathbf{b})^H = (-1)^{\mathbf{a}^T\mathbf{b}}\mathbf{D}(\mathbf{a}, \mathbf{b}). \tag{6}$$

The trace of a Pauli matrix $\mathbf{D}(\mathbf{a}, \mathbf{b})$ is

$$\mathrm{Tr}(\mathbf{D}(\mathbf{a}, \mathbf{b})) = \begin{cases} 2^n & \text{if } \mathbf{a} = \mathbf{b} = \mathbf{0} \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

meaning only the Pauli matrix $\mathbf{D}(\mathbf{0}, \mathbf{0}) = \mathbf{I}$ has trace.

The Pauli matrices can be used to form a basis for unitary matrices, meaning any $\mathbf{U} \in \mathbb{U}(2^n)$ can be written as [21]

$$\mathbf{U} = \sum_{\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n} \alpha_{\mathbf{a},\mathbf{b}} \mathbf{D}(\mathbf{a}, \mathbf{b}), \tag{8}$$

where

$$\alpha_{\mathbf{a},\mathbf{b}} = \frac{\mathrm{Tr}(\mathbf{D}(\mathbf{a}, \mathbf{b})^H \mathbf{U})}{N} \in \mathbb{C}. \tag{9}$$

Since $\mathbf{U}^H\mathbf{U} = \mathbf{I}$, we have that

$$\sum_{\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n} |\alpha_{\mathbf{a},\mathbf{b}}|^2 = 1. \tag{10}$$

Note that this is a necessary condition for the Pauli basis coordinates to give a unitary matrix, but it is not sufficient.

C. Clifford Group

The n-qubit Clifford group is defined as the set of unitaries which normalize the Pauli group $\mathcal{P}_n$, meaning

$$\mathcal{C}_n = \{\mathbf{G} \in \mathbb{U}(2^n) \mid \mathbf{G}\mathcal{P}_n\mathbf{G}^H = \mathcal{P}_n\}/\mathbb{U}(1). \tag{11}$$

The definition of $\mathcal{C}_n$ implies that for any $\mathbf{G} \in \mathcal{C}_n$ and $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n$

$$\mathbf{G}\mathbf{D}(\mathbf{a}, \mathbf{b}) = \mathbf{D}(\mathbf{c}, \mathbf{d})\mathbf{G} \tag{12}$$

for some $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^n$. The cardinality of $\mathcal{C}_n$ is given by [18]

$$|\mathcal{C}_n| = 2^{n^2+2n} \prod_{k=1}^{n} (4^k - 1). \tag{13}$$

The Hadamard matrix is denoted by $\mathbf{H}_N$ where

$$\mathbf{H}_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes n}, \tag{14}$$

where $\otimes n$ means to apply the Kronecker product n times.

D. Clifford Transvection

A Clifford transvection [21] is an operator of the form

$$\mathbf{T}_{\pm}(\mathbf{a}, \mathbf{b}) := \frac{\mathbf{I} \pm i^{(\mathbf{a}\mathbf{b}^T+1)}\mathbf{D}(\mathbf{a}, \mathbf{b})}{\sqrt{2}} \in \mathcal{C}_n. \tag{15}$$

We label the set of transvections as $\mathcal{T}_n$. A Clifford transvection is a square root of a Pauli matrix, as

$$\mathbf{T}_{\pm}^2(\mathbf{a}, \mathbf{b}) = \pm i^{(\mathbf{a}^T\mathbf{b}+1)}\mathbf{D}(\mathbf{a}, \mathbf{b}). \tag{16}$$

The inverse of a Clifford transvection is given by

$$\mathbf{T}_{\pm}^{-1}(\mathbf{a}, \mathbf{b}) = \mathbf{T}_{\pm}^{H}(\mathbf{a}, \mathbf{b}) = \mathbf{T}_{\mp}(\mathbf{a}, \mathbf{b}). \tag{17}$$

The Clifford transvections are the simplest Clifford group elements after the Pauli matrices in the sense that their Pauli basis representation in (8) contains only two non-zero terms.

E. Transvection Decomposition of the Clifford Group

Any Clifford matrix $\mathbf{G} \in \mathcal{C}_n$ is of the form

$$\mathbf{G} = \mathbf{D}(\mathbf{a}_0, \mathbf{b}_0) \prod_{j=1}^{s} \mathbf{T}_{\pm}(\mathbf{a}_j, \mathbf{b}_j), \tag{18}$$

where $\mathbf{a}_j, \mathbf{b}_j \in \mathbb{F}_2^n$ and $s \leq 2n$ [21]. Note that this decomposition is not unique.

F. Approximation Metric

To approximate a given $\mathbf{U} \in \mathbb{U}(2^n)$ with $\mathbf{G} \in \mathcal{C}_n$, we use the global phase invariant distance defined as [1]

$$d(\mathbf{G}, \mathbf{U}) = \sqrt{1 - \frac{|\mathrm{Tr}(\mathbf{G}^H\mathbf{U})|}{N}}. \tag{19}$$

This distance is global phase-invariant, since

$$|\mathrm{Tr}(a\mathbf{G}^H\mathbf{U})| = |a||\mathrm{Tr}(\mathbf{G}^H\mathbf{U})| = |\mathrm{Tr}(\mathbf{G}^H\mathbf{U})|, \tag{20}$$

for any $a \in \mathbb{C}, |a| = 1$. Note also that

$$d(\mathbf{G}, \mathbf{U}) = d(\mathbf{I}, \mathbf{G}^H\mathbf{U}). \tag{21}$$

III. Transvection-Based Clifford Approximation

For a given unitary $\mathbf{U} \in \mathbb{U}(N)$, our objective is to find

$$\widehat{\mathbf{G}} := \underset{\mathbf{G}\in\mathcal{C}_n}{\arg\min}\, d(\mathbf{G},\mathbf{U}) = \underset{\mathbf{G}\in\mathcal{C}_n}{\arg\min}\, d(\mathbf{I},\mathbf{G}^H\mathbf{U}). \tag{22}$$

The cardinality of $\mathcal{C}_n$ given in (13) grows fast with the number of qubits. For $n = 3$, $\mathcal{C}_n$ already has 2.5 million elements. With $n = 5$, there are more than 46 trillion elements. This makes an exhaustive search unfeasible. Therefore, a heuristic algorithm is needed.

The idea of our algorithm is to sequentially apply Clifford transvections $\mathbf{T} \in \mathcal{T}_n$ to $\mathbf{U}$ so that $d(\mathbf{I},\mathbf{TU})$ is minimized. In other words, we try to approach the identity matrix using Clifford transvections. Using the transvection decomposition from (18) gives

$$\mathbf{U} = \widehat{\mathbf{G}}\widetilde{\mathbf{I}} = \left(\mathbf{D}(\mathbf{a}_0,\mathbf{b}_0)\prod_{j=1}^{s}\mathbf{T}_{\pm}(\mathbf{a}_j,\mathbf{b}_j)\right)\widetilde{\mathbf{I}}, \tag{23}$$

where the error term $\widetilde{\mathbf{I}} = \widehat{\mathbf{G}}^H\mathbf{U}$ is as close to the identity matrix as we can possibly get. The hope is that our approach would eliminate the term $\widehat{\mathbf{G}}$ from (23) and leave only the term $\widetilde{\mathbf{I}}$, giving the best possible approximation.

The reason for using transvections is that finding and applying the best transvection can be done efficiently using the Pauli basis coordinates from (8). Given the Pauli basis representation

$$\mathbf{TU} = \sum_{\mathbf{a},\mathbf{b}\in\mathbb{F}_2^n}\widetilde{\alpha}_{\mathbf{a},\mathbf{b}}\mathbf{D}(\mathbf{a},\mathbf{b}), \tag{24}$$

and using (7), the distance $d(\mathbf{I},\mathbf{TU})$ can be written as

$$d(\mathbf{I},\mathbf{TU}) = \sqrt{1-\frac{|\mathrm{Tr}(\mathbf{TU})|}{N}} = \sqrt{1-|\widetilde{\alpha}_{\mathbf{0},\mathbf{0}}|}. \tag{25}$$

From (10) we see that the maximum value of $|\widetilde{\alpha}_{\mathbf{0},\mathbf{0}}|$ is 1. Therefore we define the best transvection for a unitary matrix $\mathbf{U}$ to be

$$\widehat{\mathbf{T}} := \underset{\mathbf{T}\in\mathcal{T}_n}{\arg\min}\, d(\mathbf{I},\mathbf{TU}) = \underset{\mathbf{T}\in\mathcal{T}_n}{\arg\max}\, |\widetilde{\alpha}_{\mathbf{0},\mathbf{0}}|. \tag{26}$$

Calculating $\widehat{\mathbf{T}}$ without using the Pauli basis coordinates has a complexity of $O(N^3)$. This is because one needs to calculate $|\mathrm{Tr}(\mathbf{TU})|$ for each of the $2N^2$ different Clifford transvections. Since any transvection $\mathbf{T}$ is a sum of two signed permutations, calculating $|\mathrm{Tr}(\mathbf{TU})|$ can be done in $O(N)$ operations. This gives a total computational complexity of $O(N^3)$.

However, if we have the Pauli basis coordinates of $\mathbf{U}$, finding $\widehat{\mathbf{T}}$ can be done in $O(N^2)$ operations. We are also able to calculate the Pauli basis coordinates of $\widehat{\mathbf{T}}\mathbf{U}$ in $O(N^2)$. This means that we gain an $O(N)$ speedup over the naive method, if we do all our calculations in the Pauli basis. For this, we need to calculate the Pauli basis coordinates for the input matrix $\mathbf{U}$ first. As we shall see, the Pauli basis coordinates of $\mathbf{U}$ can be computed in $O(N^2\log(N))$ operations.

A. Calculating the Pauli Basis Coordinates

For an efficient calculation of the Pauli basis coordinates, we have the following proposition.

Proposition 1. *Coordinates in the Pauli basis representation in* (8) *are given by*

$$\frac{\mathbf{H}_N}{\sqrt{N}}\,\mathrm{diag}\left(\mathbf{D}(\mathbf{x},\mathbf{0})\mathbf{U}\right) = \left[\alpha_{\mathbf{x},\mathbf{0}},\alpha_{\mathbf{x},\mathbf{b}^1},\ldots,\alpha_{\mathbf{x},\mathbf{1}}\right]^T, \tag{27}$$

where $\mathrm{diag}(\circ)$ *forms a column vector from the diagonal of its argument. Also,* $\mathbf{b}^i$ *for* $i = 1,2,\ldots,N-1$*, is the binary representation of* i *with* n *bits.*

Proof. Using Pauli basis representation of $\mathbf{U}$ in (8), we have

$$\begin{aligned}\mathbf{D}(\mathbf{x},\mathbf{0})\mathbf{U} &= \mathbf{D}(\mathbf{x},\mathbf{0})\sum_{\mathbf{a},\mathbf{b}\in\mathbb{F}_2^n}\alpha_{\mathbf{a},\mathbf{b}}\mathbf{D}(\mathbf{a},\mathbf{b})\\ &= \sum_{\mathbf{a},\mathbf{b}\in\mathbb{F}_2^n}\alpha_{\mathbf{a},\mathbf{b}}\mathbf{D}(\mathbf{a}+\mathbf{x},\mathbf{b}),\end{aligned} \tag{28}$$

where we used (5). Now $\mathbf{D}(\mathbf{a}+\mathbf{x},\mathbf{b})$ is diagonal if and only if $\mathbf{a}+\mathbf{x} = 0$, so taking the diagonal part of $\mathbf{D}(\mathbf{x},\mathbf{0})\mathbf{U}$ gives

$$\sum_{\mathbf{b}\in\mathbb{F}_2^n}\alpha_{\mathbf{x},\mathbf{b}}\mathbf{D}(\mathbf{0},\mathbf{b}). \tag{29}$$

From (3) we get that

$$\mathrm{diag}\left(\mathbf{D}(\mathbf{x},\mathbf{0})\right) = \begin{bmatrix}1\\-1\end{bmatrix}^{b_1}\otimes\cdots\otimes\begin{bmatrix}1\\-1\end{bmatrix}^{b_n} = \mathbf{h}_{\mathbf{b}}, \tag{30}$$

where $\mathbf{h}_{\mathbf{b}}$ denotes the corresponding scaled column $\mathbf{b}$ of the Hadamard matrix $\mathbf{H}_N = \frac{1}{\sqrt{N}}\left[\mathbf{h}_{\mathbf{0}},\mathbf{h}_{\mathbf{b}^1},\ldots,\mathbf{h}_{\mathbf{1}}\right]$. Therefore

$$\mathrm{diag}\left(\mathbf{D}(\mathbf{x},\mathbf{0})\mathbf{U}\right) = \sum_{\mathbf{b}\in\mathbb{F}_2^n}\alpha_{\mathbf{x},\mathbf{b}}\mathbf{h}_{\mathbf{b}}. \tag{31}$$

Applying the Hadamard transform to (31) and scaling by $\frac{1}{\sqrt{N}}$ completes the proof. $\square$

Note that $\mathbf{D}(\mathbf{x},\mathbf{0})$ is a permutation matrix. Thus finding $\mathrm{diag}(\mathbf{D}(\mathbf{x},\mathbf{0})\mathbf{U})$ is simply an indexing problem. The algorithm uses the fast Walsh-Hadamard transform to compute the distance to several codewords in parallel, as is done in the binary chirp decoding algorithm of [14]. The transform has a computational complexity of $O(N\log(N))$ [22], and it needs to be performed for all the N different $\mathbf{x}\in\mathbb{F}_2^n$. Therefore, calculating the Pauli basis has a computational complexity of $O(N^2\log(N))$.

We now have the Pauli basis coordinates of $\mathbf{U}$. From this point forward, all calculations are done using and updating these coordinates. The next two sections demonstrate how we can find and apply the best transvection $\widehat{\mathbf{T}}$ in $O(N^2)$ operations using the Pauli basis coordinates.

B. Applying a Clifford Transvection in the Pauli Basis

Let $\mathbf{T}_\pm(\mathbf{x},\mathbf{y})$ be a Clifford transvection, and $\mathbf{U} \in \mathbb{U}(N)$. Considering the Pauli basis form in (8), we have

$$\begin{aligned}\mathbf{T}_\pm(\mathbf{x},\mathbf{y})\mathbf{U} &= \left(\frac{\mathbf{I} \pm i^{(\mathbf{x}^T\mathbf{y}+1)}\mathbf{D}(\mathbf{x},\mathbf{y})}{\sqrt{2}}\right)\sum_{\mathbf{a},\mathbf{b}}\alpha_{\mathbf{a},\mathbf{b}}\mathbf{D}(\mathbf{a},\mathbf{b}) \\ &= \sum_{\mathbf{a},\mathbf{b}} \frac{\left(\alpha_{\mathbf{a},\mathbf{b}} \mp (-i)^{(\mathbf{x}^T\mathbf{y}+1)}(-1)^{(\mathbf{y}^T\mathbf{a})}\alpha_{\mathbf{a}+\mathbf{x},\mathbf{b}+\mathbf{y}}\right)}{\sqrt{2}}\mathbf{D}(\mathbf{a},\mathbf{b}).\end{aligned} \tag{32}$$

Similarly, we have

$$\begin{aligned}\mathbf{U}\mathbf{T}_\pm(\mathbf{x},\mathbf{y}) &= \sum_{\mathbf{a},\mathbf{b}}\alpha_{\mathbf{a},\mathbf{b}}\mathbf{D}(\mathbf{a},\mathbf{b})\left(\frac{\mathbf{I} \pm i^{(\mathbf{x}^T\mathbf{y}+1)}\mathbf{D}(\mathbf{x},\mathbf{y})}{\sqrt{2}}\right) \\ &= \sum_{\mathbf{a},\mathbf{b}} \frac{\left(\alpha_{\mathbf{a},\mathbf{b}} \mp (-i)^{(\mathbf{x}^T\mathbf{y}+1)}(-1)^{(\mathbf{b}^T\mathbf{x})}\alpha_{\mathbf{a}+\mathbf{x},\mathbf{b}+\mathbf{y}}\right)}{\sqrt{2}}\mathbf{D}(\mathbf{a},\mathbf{b}).\end{aligned} \tag{33}$$

This gives an $O(N^2)$ method for applying any Clifford transvection in the Pauli basis. Since any transvection $\mathbf{T}$ is a sum of two signed permutations, calculating $\mathbf{TU}$ without the Pauli basis can also be done in $O(N^2)$ operations.

C. Finding the Best Clifford Transvection

It is possible to find the best transvection faster using the Pauli basis. Let $\mathbf{T}_\pm(\mathbf{x},\mathbf{y})$ be a Clifford transvection, and $\mathbf{U} \in \mathbb{U}(N)$. Taking trace from (32) and (33) using (7) results in

$$\begin{aligned}\frac{|\mathrm{Tr}(\mathbf{T}_\pm(\mathbf{x},\mathbf{y})\mathbf{U})|}{N} &= \frac{|\mathrm{Tr}(\mathbf{U}\mathbf{T}_\pm(\mathbf{x},\mathbf{y}))|}{N} \\ &= \frac{|\alpha_{\mathbf{0},\mathbf{0}} \mp (-i)^{(\mathbf{x}^T\mathbf{y}+1)}\alpha_{\mathbf{x},\mathbf{y}}|}{\sqrt{2}}.\end{aligned} \tag{34}$$

Therefore

$$\widehat{\mathbf{T}} = \operatorname*{arg\,max}_{\mathbf{T}_\pm(\mathbf{x},\mathbf{y})} \frac{|\alpha_{\mathbf{0},\mathbf{0}} \mp (-i)^{(\mathbf{x}^T\mathbf{y}+1)}\alpha_{\mathbf{x},\mathbf{y}}|}{\sqrt{2}}. \tag{35}$$

Note that the terms $(-i)^{(\mathbf{x}^T\mathbf{y}+1)}$ can be precalculated. This gives an $O(N^2)$ method for finding the best Clifford transvection if we have the Pauli basis coordinates of $\mathbf{U}$. This is an improvement over the naive method, which takes $O(N^3)$ operations.

D. Transvection-Based Clifford Approximation Algorithm

Combining the steps above, we reach an algorithm for transvection-based Clifford approximation, summarized as Algorithm 1. The algorithm starts by calculating the Pauli basis coordinates $\alpha_{\mathbf{a},\mathbf{b}}$ for $\mathbf{U}$. All the following calculations in the algorithm are then performed using these coordinates. Then, it finds the best non-identity transvection using (35), and applies it to $\mathbf{U}$ using (32). Based on (34), applying the transvection from the left or right gives the same trace. We believe this choice to be arbitrary, and make the choice of applying transvections from the left. At some point, the algorithm begins to loop between $\mathbf{T}$ and $\mathbf{T}^{-1}$ for some $\mathbf{T} \in \mathcal{T}_n$. At this point, the algorithm stops.

Note that Algorithm 1 ignores the identity transvections when looking for the best transvection. This is because in some cases the best non-identity transvection might actually lower the absolute value of the trace, and after two steps of the algorithm, the estimation gets better. In these cases, considering the identity transvection leads to a worse approximation.

Algorithm 1 Transvection-Based Clifford Approximation

```
Input: A unitary matrix $\mathbf{U}_{\text{in}}$.
$\mathbf{G} \leftarrow \mathbf{I}$, $\mathbf{U} \leftarrow \mathbf{U}_{\text{in}}, \alpha_{\text{best}} \leftarrow 0, \mathbf{T} \leftarrow \mathbf{0}, \mathbf{T}_{\text{prev}} \leftarrow \mathbf{I}$.
Compute Pauli basis coordinates of $\mathbf{U}$ from III-A.
while $\mathbf{T} \neq \mathbf{T}_{\text{prev}}^{-1}$ do
    Find the best transvection $\mathbf{T} \in \mathcal{T}_n, \mathbf{T} \neq \mathbf{I}$ using (35).
    Use (32) and (33) to perform $\mathbf{U} \leftarrow \mathbf{TU}$ and $\mathbf{G} \leftarrow \mathbf{G}\mathbf{T}^{-1}$ in the Pauli basis.
    if $|\alpha_{\mathbf{0},\mathbf{0}}| > \alpha_{\text{best}}$ then
        $\mathbf{G}_{\text{est}} \leftarrow \mathbf{G}$, $\alpha_{\text{best}} \leftarrow |\alpha_{\mathbf{0},\mathbf{0}}|$.
    end if
    $\mathbf{T}_{\text{prev}} \leftarrow \mathbf{T}$.
end while
Output: A Clifford $\mathbf{G}_{\text{est}}$, $d(\mathbf{G}_{\text{est}}, \mathbf{U}_{\text{in}}) = \sqrt{1 - |\alpha_{\text{best}}|}$.
```

Proposition 2. *Applying Algorithm 1 to an element* $\mathbf{U} \in \mathcal{C}_n$ *of the Clifford group, always results in* $\mathbf{G}_{\text{est}} = \mathbf{U}$.

Proof. This is equivalent to Algorithm 1 applying transvections to $\mathbf{U} \in \mathcal{C}_n$ so that it transforms into the identity matrix. For some integer $s \geq 0$, the Pauli basis coordinates of $\mathbf{U}$ have 2^s non-zero elements with values $ai^k/\sqrt{2^s}$ for $k = \{0, 1, 2, 3\}$ and $a \in \mathbb{C}, |a| = 1$ [21]. Since global phase is irrelevant, we can assume that $a = 1$, and $\alpha_{\mathbf{0},\mathbf{0}} = 1/\sqrt{2^s}$.

If $\mathbf{U}$ is traceless, $\alpha_{\mathbf{0},\mathbf{0}} = 0$. Then (35) and (10) imply that there is a transvection $\mathbf{T}$ such that $\mathbf{TU}$ has a trace. Since the algorithm checks all transvections, it must find such transvection.

Suppose that $\mathbf{U}$ has a non-zero trace. In the case that $s = 0$, $\alpha_{\mathbf{0},\mathbf{0}} = 1$, which means that $\mathbf{U} = \mathbf{I}$.

Assume now that $s > 0$. Suppose that there exists a $\mathbf{T}$ such that $|\mathrm{Tr}(\mathbf{TU})| > |\mathrm{Tr}(\mathbf{U})|$. Then, the algorithm is able to find it, since it checks all possible transvections. By (34), each transvection that increases the trace will increase it by a factor of $\sqrt{2}$. Therefore, $|\mathrm{Tr}(\mathbf{TU})|/N = 1/\sqrt{2^{s-1}}$.

Suppose that there is no $\mathbf{T}$ such that $|\mathrm{Tr}(\mathbf{TU})| > |\mathrm{Tr}(\mathbf{U})|$. Then, (34) implies that for all non-zero coordinates $[\mathbf{x}, \mathbf{y}] \neq [\mathbf{0}, \mathbf{0}]$, we have $\alpha_{\mathbf{x},\mathbf{y}} = \pm i^{\mathbf{x}^T\mathbf{y}}\alpha_{\mathbf{0},\mathbf{0}} = \pm i^{\mathbf{x}^T\mathbf{y}}/\sqrt{2^s}$. This implies that $\mathbf{U}$ is Hermitian, as $(i^{\mathbf{x}^T\mathbf{y}}\mathbf{D}(\mathbf{x},\mathbf{y}))^H = i^{\mathbf{x}^T\mathbf{y}}\mathbf{D}(\mathbf{x},\mathbf{y})$. Let $[\mathbf{x}, \mathbf{y}] \neq [\mathbf{0}, \mathbf{0}]$ such that $\alpha_{\mathbf{x},\mathbf{y}} \neq 0$. Then setting $\mathbf{T} = \mathbf{T}_+(\mathbf{x},\mathbf{y}) \neq \mathbf{I}$ and using (34) gives $|\mathrm{Tr}(\mathbf{TU})| = |\mathrm{Tr}(\mathbf{U})|$. Therefore, there is a transvection that does not change the absolute value of trace. Since the algorithm checks all transvections, it must find one such transvection. Since $\mathbf{U}$ is Hermitian, the resulting matrix $\widetilde{\mathbf{U}} = \mathbf{TU}$ must have a minimal decomposition involving s transvections [21], [19]. Since

$\alpha_{\mathbf{0},\mathbf{0}} = 1/\sqrt{2^s}$, there must be a sequence of s transvections that all increase the absolute value of trace by the factor $\sqrt{2}$. Since the algorithm checks all transvections, it must find a valid transvection at each step. □

E. Regions of Suboptimality

Algorithm 1 cannot find the closest Clifford matrix for all unitary matrices. To understand why, consider the unitary operator

$$\mathbf{U} = (\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3)^a\,, \tag{36}$$

where $\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$ is the minimal transvection decomposition for a Clifford matrix $\mathbf{G}$, and $0 \le a \le 1$. The value a interpolates $\mathbf{U}$ between the identity matrix $\mathbf{I}$ ($a = 0$) and the Clifford $\mathbf{G}$ ($a = 1$). There is region of values $a \in (1/2, 1/2+\epsilon]$ for which the closest Clifford element to $\mathbf{U}$ is $\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$, but $\mathbf{U}$ is also almost as close to the identity matrix. If $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ pairwise anticommute, no product of two transvections is closer to $\mathbf{U}$ than the identity. Therefore the algorithm estimates $\mathbf{U}$ by the identity matrix, which means that we do not get the best possible Clifford estimate.

If the algorithm somehow started by applying the transvections $\mathbf{T}_1^{-1}$ and $\mathbf{T}_2^{-1}$, the algorithm would be able to get the best possible Clifford approximation, since it would find the next best transvection to be $\mathbf{T}_3^{-1}$. Therefore we have that the algorithm might give different approximation errors for $\mathbf{U}$ and $\mathbf{GU}$, where $\mathbf{G} \in \mathcal{C}_n$.

Consider an example with $\mathbf{T}_1 = \mathbf{T}_+([0,1],[0,0])$, $\mathbf{T}_2 = \mathbf{T}_+([0,1],[0,1])$ and $\mathbf{T}_3 = \mathbf{T}_+([1,0],[0,1])$. Fig. 1 shows the distances between $\mathbf{U}$ and $\mathbf{I}$, the closest transvection, the closest product of two transvections, and the matrix $\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$. Based on the figure, for values $a \in (0.5, 0.56]$, Algorithm 1 is not able to find the closest approximation for $\mathbf{U}$.

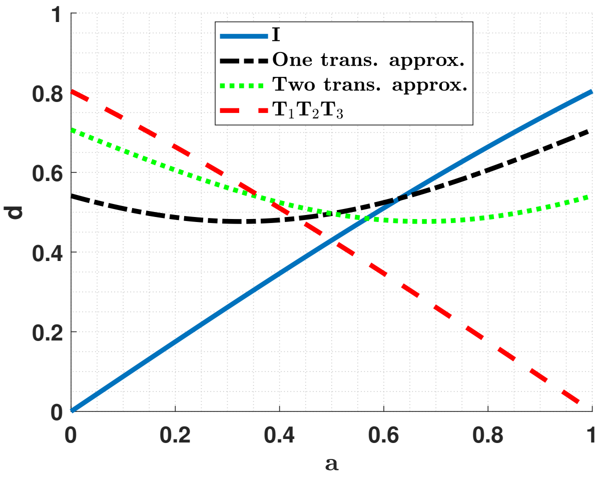

Fig. 1. Example of a region where Algorithm 1 is suboptimal. For $a \in (0.5, 0.56]$, the closest product of two transvections is always further from $\mathbf{U}$ than the identity matrix. This means that the algorithm never finds the optimal approximation $\mathbf{T}_1\mathbf{T}_2\mathbf{T}_3$.

F. Transvection-Based Approximation With Randomization

To try to circumvent the problem, randomization can be applied. For this, we choose k random transvections $\mathbf{T}_1, \cdots, \mathbf{T}_k \in \mathcal{T}_n$, and perform Algorithm 1 for $\mathbf{U}_0 = \mathbf{U}$, and for the matrices $\mathbf{U}_j = \left(\prod_{i=1}^{j} \mathbf{T}_i\right)\mathbf{U}$, where $1 \le j \le k$. The algorithm then chooses the best approximation from these $k+1$ options. The intuition is that the randomized algorithm would efficiently explore different starting points for the algorithm. The randomized algorithm is presented in Algorithm 2.

Algorithm 2 Transvection-Based Clifford Approximation With Randomization

```
Input: A unitary matrix $\mathbf{U}_{\text{in}}$, $k \in \mathbb{Z}_+$.
$\mathbf{U}_0 \leftarrow \mathbf{U}_{\text{in}}$.
Compute the Pauli basis of $\mathbf{U}_0$ as described in III-A.
Compute $\mathbf{G}_0$ and $d(\mathbf{G}_0, \mathbf{U}_0)$ using Algorithm 1 on $\mathbf{U}_0$.
$d_{\text{best}} \leftarrow d(\mathbf{G}_0, \mathbf{U}_0)$, $j_{\text{best}} \leftarrow 0$
Select $k$ random Clifford transvections $\mathbf{T}_1, ..., \mathbf{T}_k$.
for $j = 1$ to $k$ do
    $\mathbf{U}_j \leftarrow \mathbf{T}_j\mathbf{U}_{j-1}$
    Compute $\mathbf{G}_j$ and $d(\mathbf{G}_j, \mathbf{U}_j)$ using Algorithm 1 on $\mathbf{U}_j$.
    if $d(\mathbf{G}_j, \mathbf{U}_j) < d_{\text{best}}$ then
        $d_{\text{best}} \leftarrow d(\mathbf{G}_j, \mathbf{U}_j)$, $j_{\text{best}} \leftarrow j$.
    end if
end for
$\mathbf{G}_{\text{est}} = \mathbf{T}_1^{-1} \cdots \mathbf{T}_{j_{\text{best}}}^{-1}\mathbf{G}_{j_{\text{best}}}$.
Output: A Clifford $\mathbf{G}_{\text{est}}$, $d(\mathbf{G}_{\text{est}}, \mathbf{U}_{\text{in}}) = d_{\text{best}}$.
```

G. Complexity Analysis

As discussed in sections III-A, III-B and III-C, calculating the Pauli basis has a computational complexity $O(N^2\log(N))$, and finding and applying the best transvection has complexity $O(N^2)$. Since the minimal transvection decomposition for any Clifford has at most $2n$ transvections [16], the number of transvections applied in the algorithm is proportional to $n = \log(N)$. Therefore, the computational complexity of the Algorithm 1 is $O(N^2\log(N))$, or $O(n\,4^n)$ in terms of the number of qubits. With the Algorithm 2, the base algorithm is used k times, giving an overall complexity $O(kN^2\log(N))$.

IV. NUMERICAL RESULTS

In this section, we compare the Transvection-Based Clifford Approximation (TBCA) algorithm and its randomized version with a method based on the block-ZXZ decomposition [7]. The block-ZXZ algorithm decomposes a unitary matrix into CNOT gates and arbitrary single qubit gates. To find a Clifford approximation, we perform exhaustive search for approximating the single qubit operations with Clifford operations.

For each number of qubits n, we generated n_s random unitary matrices using the method described in [23], and calculated the root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s} d^2(\mathbf{G}_{\text{est},i}, \mathbf{U}_i)}. \tag{37}$$

Fig. 2 illustrates the simulation results for 1-RMSE, with $n_s = 1000$ samples for each number of qubits in the range $n = 1, \ldots, 8$. In this figure, TBCA, Randomization and Baseline are referring to Algorithm 1, Algorithm 2, and the modified block-ZXZ algorithm, respectively. TBCA was implemented in Matlab, and the Baseline was implemented in Python.

Note that the maximum distance is 1, and due to the sparseness of Clifford matrices, the RMSE is expected to approach 1 as n increases. The results show that TBCA is considerably better at approximating a unitary matrix in the set of Clifford-matrices than the block-ZXZ algorithm. Also, we observe that randomization is only slightly improving TBCA performance. One reason for this can be seen in Figure 1. In the presented case, randomized TBCA can perform better only in the suboptimality region $a \in (0.5, 0.56]$, and in that region, the possible improvements are relatively small.

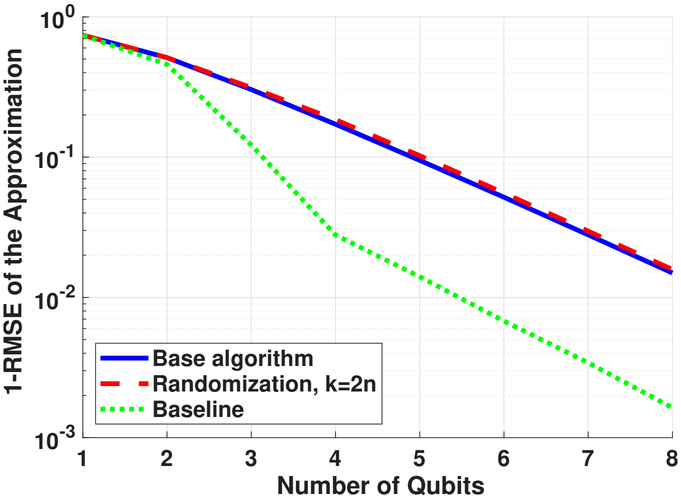

Fig. 2. Comparing 1-RMSE for the different algorithms. For Randomization we used $k = 2n$.

V. CONCLUSION AND FUTURE WORK

We developed algorithms for finding the closest Clifford-group element to an arbitrary unitary matrix, inspired by algorithms for decoding algebraically defined vector codebooks. The algorithms showed promising performance, when compared to a state of the art method from the literature. As a future work, we suggest to extend this approach to higher levels of the Clifford hierarchy. For example, for the third level, one could use square roots of Clifford transvections. However, more care is required, as the third level of the Clifford hierarchy does not form a group. This means that multiplying square roots of Clifford transvections might not result in a Clifford hierarchy element. Hence, a more suitable first step would be to extend this algorithm to higher levels of the diagonal Clifford hierarchy, since it forms a group [13].

ACKNOWLEDGEMENTS

This work was supported in part by the Finnish Ministry of Education and Culture through the Quantum Doctoral Education Pilot Program (QDOC, VN/3137/2024-OKM-4), and the Research Council of Finland through the Finnish Quantum Flagship Project under Grant 358877, and in part by Business Finland under Grant 8264/31/2022.

REFERENCES

[1] V. Kliuchnikov, D. Maslov, and M. Mosca, “Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits,” *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 161–172, 2015.

[2] P. Selinger, “Efficient Clifford+T approximation of single-qubit operators,” *Quantum Inf. Comput.*, vol. 15, no. 1-2, pp. 159–180, Jan. 2015.

[3] N. Ross and P. Selinger, “Optimal ancilla-free Clifford+T approximation of z-rotations,” *Quantum Inf. Comput.*, vol. 16, no. 11-12, pp. 901–953, Sep. 2016.

[4] M. Amy and M. Mosca, “T-count optimization and reed–muller codes,” *IEEE Trans. Inform. Theory*, vol. 65, no. 8, pp. 4771–4784, 2019.

[5] A. Barenco & al., “Elementary gates for quantum computation,” *Phys. Rev. A*, vol. 52, p. 3457–3467, 1995.

[6] M. Möttönen and J. J. Vartiainen, “Decompositions of general quantum gates,” in *Trends in Quantum Computing Research*, 2006, pp. 149–170.

[7] A. M. Krol and Z. Al-Ars, “Beyond quantum shannon decomposition: Circuit construction for n-qubit gates based on block-zxz decomposition,” *Physical Review Applied*, vol. 22, no. 3, p. 034019, 2024.

[8] V. Kliuchnikov, D. Maslov, and M. Mosca, “Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates,” *Quantum Inf. Comput.*, vol. 13, no. 7-8, pp. 607–630, 2013.

[9] A. Krol, A. Sarkar, I. Ashraf, Z. Al-Ars, and K. Bertels, “Efficient decomposition of unitary matrices in quantum circuit compilers,” *Appl. Sci.*, vol. 12, no. 759, pp. 1–20, 2022.

[10] D. Gottesman and I. L. Chuang, “Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations,” *Nature*, vol. 402, no. 6760, pp. 390–393, 1999.

[11] B. Zeng, X. Chen, and I. L. Chuang, “Semi-Clifford operations, structure of $\mathcal{C}_k$ hierarchy, and gate complexity for fault-tolerant quantum computation,” *Physical Review A—Atomic, Molecular, and Optical Physics*, vol. 77, no. 4, p. 042313, 2008.

[12] N. de Beaudrap, X. Bian, and Q. Wang, “Fast and effective techniques for T-count reduction via spider nest identities,” in *Proc. Conf. Theory of Quantum Comput., Commun. and Crypto.*, no. 11, 2020, pp. 1–23.

[13] S. X. Cui, D. Gottesman, and A. Krishna, “Diagonal gates in the clifford hierarchy,” *Physical Review A*, vol. 95, no. 1, p. 012329, 2017.

[14] S. D. Howard, A. R. Calderbank, and S. J. Searle, “A fast reconstruction algorithm for deterministic compressive sensing using second order Reed-Muller codes,” in *Conference on Information Sciences and Systems*, March 2008, pp. 11–15.

[15] R. Calderbank, S. Howard, and S. Jafarpour, “A sublinear algorithm for sparse reconstruction with ℓ_2/ℓ_2 recovery guarantees,” in *IEEE Internat. Workshop on Comput. Advances in Multi-Sensor Adapt. Proc. (CAMSAP)*, 2009, pp. 209–212.

[16] T. Pllaha, O. Tirkkonen, and R. Calderbank, “Binary subspace chirps,” *IEEE Trans. Inform. Theory*, no. 12, pp. 7735–7752, Dec. 2022.

[17] M. Bayanifar, E. Heikkilä, R. Calderbank, and O. Tirkkonen, “Extended binary chirps codebooks for non-coherent communications,” in *IEEE GLOBECOM*, 2023, pp. 1824–1829.

[18] R. Koenig and J. A. Smolin, “How to efficiently select an arbitrary Clifford group element,” *Journal of Mathematical Physics*, vol. 55, no. 12, 2014.

[19] T. Pllaha, K. Volanto, and O. Tirkkonen, “Decomposition of Clifford gates,” in *Proc. IEEE Globecom*, Dec. 2021, pp. 1–6.

[20] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[21] T. Pllaha, N. Rengaswamy, O. Tirkkonen, and R. Calderbank, “Un-Weyling the Clifford hierarchy,” *Quantum*, vol. 4, p. 370, 2020.

[22] Fino and Algazi, “Unified matrix treatment of the fast Walsh-Hadamard transform,” *IEEE Trans. Comput.*, vol. 100, no. 11, pp. 1142–1146, 1976.

[23] F. Mezzadri, “How to generate random matrices from the classical compact groups,” *Notices of the American Mathematical Society*, vol. 54, no. 5, pp. 592 – 604, May 2007.