Two-Qubit Gate Synthesis via Linear Programming for Heterogeneous Instruction Sets

Evan McKinney*[†], Lev S. Bishop[†]

*Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA

[†]IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

evm33@pitt.edu, lsbishop@us.ibm.com

Abstract—We present GULPS (Global Unitary Linear Programming Synthesis), a segmented Cartan trajectory method that compiles arbitrary two-qubit unitaries into native gate sets by decomposing the synthesis problem into locally solvable sub-problems. Each segment corresponds to a depth-2 circuit synthesized from a linear program over canonical gate invariants, subject to quantum Littlewood-Richardson (QLR) constraints. The intermediate invariants are stitched together via nonlinear least-squares optimization to recover the local operations between segments. This hybrid LP-numerical method enables robust synthesis across parameterized instruction sets. As quantum hardware continues to evolve, GULPS provides a scalable, ISA-aware compilation strategy suitable for integration into platforms like Qiskit.

Index Terms—Quantum computing, quantum compilation

I. INTRODUCTION

Two-qubit interactions are fundamental to quantum circuits. Beyond generating entanglement, they are crucial for implementing a wide array of quantum algorithms, decomposing high-level unitaries, and mapping these decompositions onto hardware via SWAP operations. For example, variational algorithms exploit two-qubit operations to simulate two-body Hamiltonian terms, while algorithms such as Shor's and Grover's eventually decompose into sequences of controlled-phase two-qubit gates through methods like the Cosine-Sine or Quantum Shannon decompositions [1]–[4].

Recent advancements in hardware have led to the emergence of increasingly heterogeneous quantum instruction set architectures (ISA). Modern superconducting qubit platforms routinely implement multiple types of two-qubit interactions, including iSWAP (mediated by microwave photon exchange) and CZ (phase accumulation on the $|f\rangle$ state) [5]. Advanced techniques such as frame tracking have enabled standard cross-resonance gates to be transformed into general SU(4) operators, substantially widening the set of available gate primitives [6]. In addition, the development of fractional [7], [8] and continuous gates [9]–[12] provide clear evidence of the progress in hardware capabilities.

Historically, compilation and synthesis have been framed primarily in terms of the CNOT gate—a convention that worked well when systems were built around CZ or cross-resonance basis gates. Despite the increasing hardware diversity, synthesis techniques have largely remained confined to this CNOT-centric view [13], [14]. While analytical compilation methods exist for specific families of gates (such as the
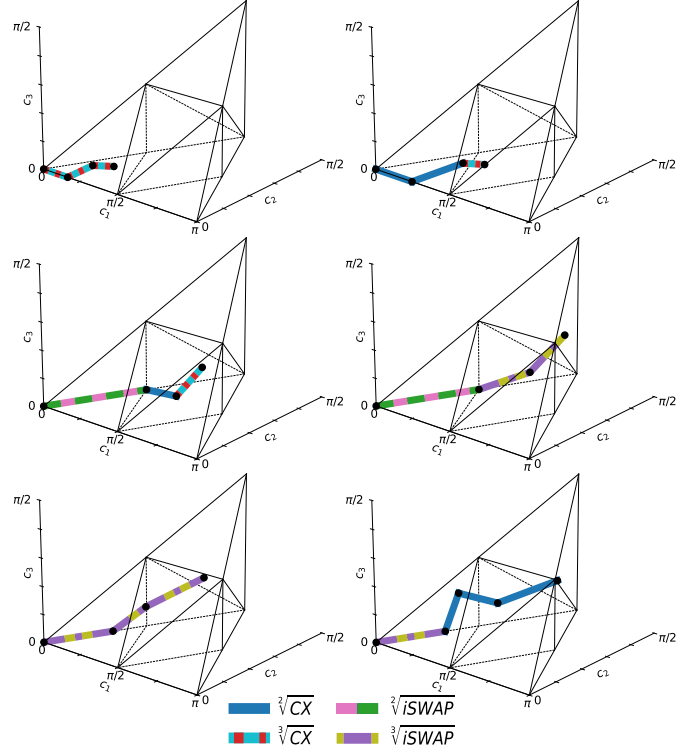

Fig. 1. Segmented trajectory decompositions in the Weyl chamber produced by the decomposer for the ISA $\mathcal{G} = \{\sqrt[2]{CX}, \sqrt[3]{CX}, \sqrt[2]{iSWAP}, \sqrt[3]{iSWAP}\}$.

B-gate, the XX-family, iSWAP, and $\sqrt{\text{iSWAP}}$), they do not fully exploit the potential of a heterogeneous set of two-qubit interactions. Access to a diverse basis not only reduces the number of operations required for circuit synthesis but also enables smaller gate angles to lower overall decomposition costs, more flexible parameterized circuit (PQC) ansätze for variational problems, and more advanced circuit routing strategies [15]–[18].

Current numerical synthesis methods can handle these varied bases to some extent; however, they will ultimately struggle with scalability as the instruction sets become increasingly heterogeneous [19]–[22]. This limitation highlights the urgent need for novel approaches that can efficiently leverage the full ISAs available in modern quantum hardware architectures.

In this work, we introduce **GULPS**, a robust compiler

1

technique that scales to arbitrarily complex instruction sets by breaking a large, intractable optimization into a series of smaller, independently solvable subproblems. Our synthesis approach views the implementation of a target two-qubit unitary as traversing a trajectory in the space of gate invariants that move from the identity to the target unitary. This is motivated by continuous parameterizations of gate invariants, described by a Cartan "flow" using a time-dependent control Hamiltonian [23]. Rather than attempting to solve the global synthesis problem in one shot, GULPS decomposes it into a sequence of depth-two segments, each synthesized using a linear program with appropriate basis gates chosen from the ISA.

Key contributions of this work include:

- **Segmented Cartan Trajectory Formulation:** We propose a novel formulation that decomposes the two-qubit gate synthesis problem into independently solvable, parameterized depth-two circuits.
- **LP-Based Intermediate Invariants Solver:** We develop an efficient linear programming (LP)-based solver by reformulating monodromy polytope constraints as sets of linear inequalities.
- **Numerical Recombination Strategy:** We introduce a robust numerical method to recombine the independently optimized segments into a complete decomposition compatible with the target quantum instruction set.

II. BACKGROUND

Two-qubit gate invariants have been extensively studied and provide a concise means to characterize the nonlocal properties of two-qubit unitaries. In particular, every two-qubit unitary operation $U \in \mathrm{PU}(4)$ (*i.e.* the unitary group up to a global phase) can be expressed in a canonical form via the Cartan KAK decomposition [24]. In this form, the nonlocal content of U is separated from the local single-qubit rotations:

$$U = K \cdot \mathrm{CAN}(a_1, a_2, a_3) \cdot K', \tag{1}$$
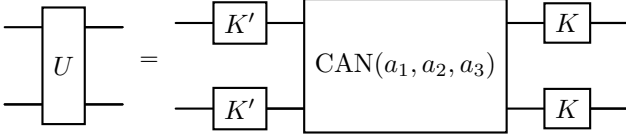
where $K, K' \in \mathrm{PU}(2) \times \mathrm{PU}(2)$ are local unitaries, and

$$\mathrm{CAN}(a_1, a_2, a_3) = \exp\left(-i\left(a_1 XX + a_2 YY + a_3 ZZ\right)\right) \tag{2}$$

is the canonical two-qubit gate. The three real parameters (a_1, a_2, a_3) serve as invariants that completely characterize the nonlocal properties of U [23]–[25].

Two-qubit gates are defined to be *locally equivalent* if they share the same canonical parameters—that is, if there exist local unitaries K, K' such that

$$U = K V K' \implies U \simeq V.$$

In practice, one can promote local equivalence to a stronger unitary equivalence when needed. For example, expressing

$$\mathrm{CAN}(U) = K U K' \quad \text{and} \quad \mathrm{CAN}(V) = L V L',$$

if $\mathrm{CAN}(U) = \mathrm{CAN}(V)$, then we have

$$U = K^\dagger L V L' K'^\dagger. \tag{3}$$

Gate invariants provide a concise way to characterize the nonlocal properties of two-qubit gates. They are particularly useful for guiding unitary synthesis or optimal control derived from physical Hamiltonians, *e.g.* relating measures of fidelity to geometric distances [26]–[33]. Moreover, since single-qubit gates (and global phases) incur minimal cost (e.g., virtual-z rotations) [34], they can be set aside when focusing on the characteristics of the ISA.

Remark II.1. *Various representations of two-qubit invariants exist, but they all carry the same information and can be converted between each other using explicit formulas [35]. The monodromy inequalities use the logarithmic spectrum of the Cartan double $(\gamma(U^Q))$. However, we later translate into Makhlin invariants to write a differentiable cost function and the Weyl chamber invariants for visualization.*

A. The monodromy polytope linear inequalities

In previous works [36], it has been shown that the subspace of two-qubit unitaries reachable by a circuit containing two two-qubit gates can be completely characterized by a finite set of linear inequalities. This formulation reduces the synthesis problem to a *multiplicative eigenvalue problem*: if one considers two fixed two-qubit gates, say G_1 and G_2, then any unitary T obtained by

$$T = K_2 G_2 K_1 G_1 K_0$$

is constrained to a certain range of canonical parameters. This is given by the key result (Theorem 23, [36]): if one selects integers $r, k > 0$ with $r + k = n$ (with $n = 4$ in the two-qubit case) and chooses sequences $a \in Q_{r,k}, \quad b \in Q_{r,k}, \quad c \in Q_{r,k}$, such that the associated quantum Littlewood–Richardson coefficient $N_{a,b}^{c,d}(r, k) = 1$, then the canonical parameters must satisfy the linear inequality

$$d - \sum_{i=1}^{r}\left(\alpha_{k+i} - a_i\right) - \sum_{i=1}^{r}\left(\beta_{k+i} - b_i\right) + \sum_{i=1}^{r}\left(\delta_{k+i} - c_i\right) \geqslant 0. \tag{4}$$

The collection of all such QLR inequalities taken over the allowed choices of r, k, and sequences from $Q_{r,k}$ defines the *monodromy polytope*, that is, the convex region in two-qubit invariant space.

B. Circuit sentences

Let $\mathcal{G}$ denote the two-qubit ISA, which—in principle—can be parameterized; here, however, we assume that $\mathcal{G}$ is a discrete set of two-qubit basis gates. A *circuit sentence* built on this ISA is a fixed sequence of these basis gates interleaved

with local single-qubit rotations. Formally, a circuit sentence can be expressed as

$$S : \theta \mapsto \Big(K_0(\theta),\, G_1,\, K_1(\theta),\, G_2,\, \ldots,\, G_n,\, K_n(\theta) \Big), \quad (5)$$

where each $G_j \in \mathcal{G}$ is a fixed basis gate and each $K_j(\theta)$ is a local unitary that depends on the parameter θ. The set of all unitaries achievable by varying the local rotations for a fixed basis gate sequence forms a convex region in invariant space; this region is referred to as the *circuit polytope*. While the local rotations determine the specific final unitary produced by the circuit sentence, they do not affect the span of nonlocal invariants conditioned by the fixed sequence $(G_1, G_2, \ldots, G_n)$. Consequently, the sentence uniquely determines the circuit polytope. To emphasize the invariance of the circuit polytope with respect to the ordering of fixed basis gates, consider the following result.

Theorem II.1. *For any circuit sentence S and for any permutation π of the fixed gates $(G_1, \ldots, G_n)$, there exist local unitaries such that the reordered sentence*

$$\tilde{S} : \theta \mapsto \Big(\tilde{K}_0(\theta),\, G_{\pi(1)},\, \tilde{K}_1(\theta),\, G_{\pi(2)},\, \ldots,\, G_{\pi(n)},\, \tilde{K}_n(\theta) \Big)$$

implements a unitary that is locally equivalent to that produced by S. In other words, the circuit polytope is invariant under any permutation of the fixed gates.

Proof. We first show that an adjacent swap can be compensated by appropriate local unitaries. Consider the two-gate circuit sentence

$$T = K_2\, G_2\, K_1\, G_1\, K_0.$$

Taking the adjoint of the entire circuit yields

$$T^\dagger = K_0^\dagger\, G_1^\dagger\, K_1^\dagger\, G_2^\dagger\, K_2^\dagger.$$

We assert that for any fixed gate U, we have $U^\dagger \sim U$, with "$\sim$" indicating a weaker local equivalence—the invariants are preserved up to sign changes that do not affect the overall solution space. Since the adjoint of a local unitary remains a local unitary, it follows that

$$T^\dagger \sim K_0^\dagger\, G_1\, K_1^\dagger\, G_2\, K_2^\dagger.$$

so by relabeling the local unitaries (e.g., $\tilde{K}_2 = K_0^\dagger$, $\tilde{K}_1 = K_1^\dagger$, $\tilde{K}_0 = K_2^\dagger$), we find

$$T \sim \tilde{K}_2\, G_1\, \tilde{K}_1\, G_2\, \tilde{K}_0.$$

Since any permutation of the fixed gates in a circuit sentence can be achieved by a sequence of such adjacent swaps, the argument extends by induction to the general case. Hence, reordering the fixed gates leaves the overall nonlocal operation (and thus the circuit polytope) invariant. $\square$
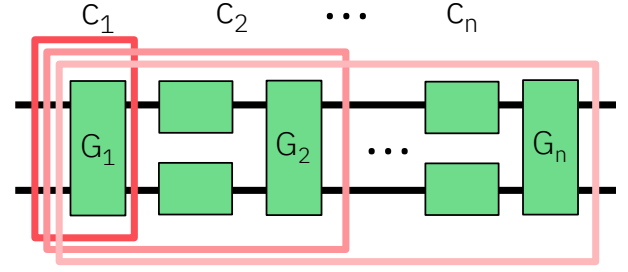

Fig. 2. Circuit invariants for each basis gate step for some two-qubit decomposition circuit sentence.

III. SEGMENTED CARTAN TRAJECTORIES

GULPS constructs a complete synthesis trajectory—from the identity to a target two-qubit gate—by partitioning the decomposition into a sequence of contiguous segments, each constrained by the QLR inequalities. The set of inequalities governing the ith decomposition step is expressed as

$$\mathcal{L}(C_{i-1}, G_i, C_i) \leqslant 0, \quad \text{for } i = 1, \ldots, n, \qquad (6)$$

where C_{i-1} is the canonical invariant (a triplet of real numbers) immediately preceding the application of the basis gate with invariant G_i, and C_i is the invariant that results after its application (Fig. 2).

A complete and valid decomposition is achieved when the union of all these segment constraints,

$$\bigcup_{i=1}^{n} \mathcal{L}(C_{i-1}, G_i, C_i) \leqslant 0, \qquad (7)$$

is simultaneously satisfied along with the prescribed boundary conditions

$$\mathrm{CAN}(C_0) \simeq I \quad \text{and} \quad \mathrm{CAN}(C_n) \simeq T, \qquad (8)$$

where I is the identity and T is the target unitary.

Since C_1 is determined directly by the first basis gate G_1, the constraints corresponding to the first two steps, $\mathcal{L}(C_0, G_1, C_1)$ and $\mathcal{L}(C_1, G_2, C_2)$, can be merged into a single effective constraint $\mathcal{L}(G_1, G_2, C_2) \leqslant 0$. In a sequence involving n basis gates, the overall formulation comprises $72(n-1)$ linear inequalities (since the segment from C_0 to G_1 is trivial) and introduces $3(n-2)$ free continuous variables (the components of $C_2, \ldots, C_{n-1}$, with C_0 and C_n fixed).

A. Linear programming formalism

GULPS reduces the gate synthesis problem to a feasibility problem over the intermediate canonical invariants $C_1, \ldots, C_{n-1}$. We collect the monodromy constraints into a standard linear program (LP) of the form $Ax \leqslant b$, where the decision vector x comprises all free invariants and other auxiliary variables associated with gate selection or parameterization. We consider four such formulations:

- **Fixed Sentence, Discrete ISA.**
  In this case, each basis gate G_i is selected from a known discrete set $\mathcal{G}$, and the complete sentence $(G_1, \ldots, G_n)$

is specified in advance. The LP then solves solely for the intermediate invariants:

$$x^T = [C_2, C_3, \ldots, C_{n-1}] \in \mathbb{R}^{3(n-2)}. \qquad (9)$$

The objective is trivial—merely to find any feasible solution:

$$\min 0 \quad \text{subject to} \quad Ax \leqslant b. \qquad (10)$$

Candidate sentences are ordered by a total cost heuristic (see Theorem II.1), and LP feasibility is checked for each.

- **Fixed Sentence, Parameterized ISA.**
  Here the sentence structure is fixed, but each G_i is drawn from a continuously parameterized gate family (*e.g*, $XY(\phi)$). The LP now solves for both the gate parameters and the invariants:

  $$x^T = [\phi_1, \ldots, \phi_n, C_2, \ldots, C_{n-1}], \qquad (11)$$

  with an objective such as minimizing the total interaction strength:

  $$\min \sum_i \phi_i \quad \text{subject to} \quad Ax \leqslant b. \qquad (12)$$

- **Gate Selection via MILP, Discrete ISA.**
  If the sentence is not predetermined, GULPS can be extended to choose each gate from the ISA $\mathcal{G} = \{G^{(1)}, \ldots, G^{(m)}\}$ using binary variables $k_{ij} \in \{0,1\}$, where $k_{ij} = 1$ indicates that gate $G^{(j)}$ is selected at position i. The decision vector then:

  $$x^T = [k_{11}, \ldots, k_{nm}, C_2, \ldots, C_{n-1}]. \qquad (13)$$

  Subject to the constraint that for each position i:

  $$\sum_{j=1}^m k_{ij} = 1, \quad \forall i. \qquad (14)$$

  This yields a mixed-integer LP (MILP), which incurs significant overhead compared to enumerating fixed sentences.

- **Gate Selection, Parameterized ISA.**
  In the most general formulation, both binary selection variables k_{ij} and continuous gate parameters $\phi_i^{(j)}$ are introduced, leading to conditional constraints:

  $$k_{ij} = 1 \Rightarrow G_i = G^{(j)}(\phi^{(j)}). \qquad (15)$$

  Although this supports extremely rich ISAs, it is unlikely to be practical without aggressive constraint pruning or hybrid search strategies.

In this work, we focus primarily on the **discrete ISA with fixed sentence** setting. For each decomposition segment, the constraints are expressed as

$$\mathcal{L}_j(C_{i-1}, G_i, C_i) := A_j^{(i)} x \leqslant b_j^{(i)}, \qquad (16)$$

where $A_j^{(i)}$ targets the relevant subvector of decision variables x and fixed components (such as C_0, C_n, and gate invariants) are incorporated into the right-hand side b. The full system is constructed by stacking these constraint blocks across all segments. Other LP variants maintain essentially the same block structure but require more complex indexing.

For ISAs of moderate size ($|\mathcal{G}| \lessgtr 10$), we find that it is significantly more efficient to enumerate gate sequences and solve a fully continuous LP for each candidate sentence. By ordering sentences using a cost-prioritized queue based on interaction strength, gate duration, or expected infidelity, GULPS ensures that the first feasible decomposition encountered is also the cheapest. This strategy yields rapid rejection of infeasible sequences while exploiting the efficiency of LP solvers.

Although we implemented the full MILP-based formulation, the overhead introduced by binary decision variables was substantial, even when including specialized optimizations such as symmetry breaking. In practice, the MILP formulation becomes less attractive especially when the optimal sentence length is unknown in advance. In such cases, the solver must assume a maximum sentence length and pad unused slots with identity gates, further increasing the variable count and problem complexity.

For most purposes, there is no observable difference between T and $-T$. Thus, GULPS considers both the target and its ρ-reflection. In practice, one can either solve two distinct candidate LPs sequentially, one for T and one for $-T$, or in the MILP formulation, introduce an additional binary variable to select between T and $-T$ (see Corollary 25 [36]).

Therefore, the LP step in GULPS determines the intermediate coordinates $C_1, \ldots, C_{n-1}$. The next step is to compute the local operations that interleave each basis gate, thereby synthesizing each segment of the overall circuit sentence.

B. Numerical synthesis per segment

Once the LP solver returns the sequence of intermediate canonical invariants $\{C_i\}$ and a basis gate sequence $\{G_i\}$, the remaining task is to synthesize the local rotations that stitch these segments together. In this stage, our goal is to compute the unknown interior local rotations, denoted by U, that when inserted between the canonical invariant C_{i-1} and the next basis gate G_i, yield a unitary locally equivalent to the desired invariant. The corresponding exterior local rotations, denoted by K, are then recovered via the Cartan KAK decomposition to upgrade local equivalence to full unitary equivalence. Here, U and K belong to $\mathrm{PU}(2) \times \mathrm{PU}(2)$; for brevity, we do not index K even though different segments use distinct exterior rotations.

Consider a three-segment decomposition (omitting the trivial initial segment). The LP yields:

$$C_2 \simeq G_2 U_1 G_1; \quad T \simeq G_3 U_2 C_2.$$

Thus, there exists K such that

$$\mathrm{CAN}(C_2) = K \left(G_2 U_1 G_1 \right) K.$$

Since C_2 is defined solely by its invariants, we can represent it in canonical form, so that

$$T \simeq G_3 U_2 \mathrm{CAN}(C_2).$$

Once U_1 and U_2 are determined, the overall unitary is assembled by recovering the missing exterior rotations. In the three-segment case, the final unitary can be written as

$$T = K\Big[\big(K\,G_3\,U_2\,G_2\,K\big)\,U_1\,G_1\Big]K,$$

which, after merging adjacent local rotations, assumes the form

$$T = K\,G_3\,K\,G_2\,K\,G_1\,K.$$

More generally, given an n-segment LP solution, the synthesis problem reduces to independently solving, for each segment i:
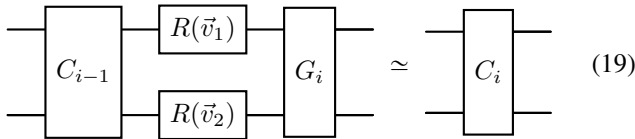
$$\mathrm{CAN}(C_i) \simeq \mathrm{CAN}(C_{i-1})\,U_{i-1}\,G_i, \qquad (17)$$

which corresponds to the effective saturation problem (Problem 12.2 in [36]). After computing the local gates, the final target unitary is then expressed in a nested form:

$$T = K\Big(G_n\,U_{n-1}\Big(\cdots\Big(K\,G_2\,U_1\,G_1\Big)\cdots\Big)\Big)K. \qquad (18)$$

Note that the synthesis can, in principle, be performed sequentially. Once the interior rotation U_i and its exterior recovery are computed, the synthesized unitary for that segment may be substituted into the next segment's template, eliminating the need for nested recovery operations. In our implementation, however, we solve each segment independently, as the overhead of merging single-qubit gates is negligible and parallelizing subproblems offers practical benefits.

In our implementation, we parameterize the interior rotation as an RV gate. Specifically, we associate two 3-parameter vectors, $\vec{v}_1, \vec{v}_2 \in \mathbb{R}^3$, with the single-qubit rotations applied on each qubit. This leads to the following ansatz for a given segment:

$$(19)$$

where the rotation is defined as

$$R(\vec{v}) = \exp\Big(-i\,\frac{\vec{v}\cdot\sigma}{2}\Big), \qquad (20)$$

with $\sigma = (\sigma_x, \sigma_y, \sigma_z)$ the Pauli matrices.

To quantify the agreement between the candidate unitary and the target, we compare their Makhlin invariants $M[G]$. We define the residual as

$$r(\vec{v}_1, \vec{v}_2) = M\Big[U(\vec{v}_1, \vec{v}_2)\Big] - M\big[U_{\text{target}}\big], \qquad (21)$$

and our objective is to minimize the least-squares cost function:

$$\min_{\vec{v}_1,\vec{v}_2\in\mathbb{R}^3} \|r(\vec{v}_1, \vec{v}_2)\|^2. \qquad (22)$$

An optimal solution with $r = 0$ is guaranteed to exist.

This optimization is performed using the Levenberg–Marquardt algorithm, with the Jacobian of $r(\vec{v}_1, \vec{v}_2)$ computed via automatic differentiation. The initial guesses for

$\vec{v}_1$ and $\vec{v}_2$ are sampled uniformly from $[-2\pi, 2\pi]^3$, and the algorithm iterates until the residual norm falls below a preset tolerance or a maximum number of iterations is reached.

Once the optimal parameters, $\vec{v}_1^*$ and $\vec{v}_2^*$, are obtained, they are converted into circuit operations implementing the interior rotation U. The missing exterior rotations are then recovered using the Cartan KAK decomposition, yielding the final unitary for the segment:

$$\mathrm{CAN}(C_i) = K\,\Big(G_i\,\big(R(\vec{v}_1^*)\otimes R(\vec{v}_2^*)\big)\,C_{i-1}\Big)K. \qquad (23)$$

By applying this procedure independently to each segment, the synthesis of complex two-qubit unitaries is reduced to a series of robust, low-dimensional optimization problems.

IV. Results and Discussion

We first verify that our segmented synthesis procedure, GULPS, faithfully implements the desired target two-qubit unitaries. For each test case, we multiply out the synthesized circuit and compare the resulting unitary to the original target, confirming agreement up to local equivalence. To visualize the decomposition process, we sample Haar-random targets [37] and plot the corresponding segmented Cartan trajectories in the Weyl chamber (Fig. 1). These visualizations demonstrate that the trajectory proceeds contiguously from the identity to the target, validating both the intermediate invariants returned by the LP and the accuracy of the numerical stitching.

To make synthesis cost-aware, we associate a relative duration cost to each basis gate. In this experiment, we assume that a CX and a $\sqrt{iSWAP}$ gate incur equal time on the target device—though other hardware may assign different costs. Accordingly, we assign the following relative durations to the basis gates:

$$\mathcal{G} = \left\{ \sqrt[2]{CX} : \frac{1}{2},\ \sqrt[3]{CX} : \frac{1}{3},\ \sqrt[2]{iSWAP} : 1,\ \sqrt[3]{iSWAP} : \frac{2}{3} \right\}.$$

These factors are incorporated into decomposition during iteration of candidate sentences, in increasing order of cost: $S_1 = \{\sqrt[3]{CX}\}, S_2 = \{\sqrt[2]{CX}\}, S_3 = \{\sqrt[2]{iSWAP}\}, S_4 = \{\sqrt[3]{CX}, \sqrt[3]{CX}\}, S_5 = \{\sqrt[2]{CX}, \sqrt[3]{CX}\}$, and so on. This ordering ensures that the first feasible trajectory found by the LP is the cheapest.

Next, to evaluate synthesis performance, we benchmarked the time required to determine a valid circuit sentence using GULPS. Figure 3 shows the distribution of times required to find the first feasible sentence for 10,000 Haar-random target gates using the instruction set

$$\mathcal{G} = \{CX, \sqrt[2]{CX}, \sqrt[3]{CX}\}.$$

This set matches the default ISA for Qiskit's `XXDecomposer` (v1.4.0). GULPS is nearly as fast as Qiskit's approach (within a factor of 3), even though it is more general and not limited to XX-type basis gates. (Note that this comparison evaluates only the determination of `_best_decomposition()`, excluding the synthesis of local gate rotations.) The distribution of solution times is not unimodal, reflecting the fact that different target gates require varying numbers of rejected
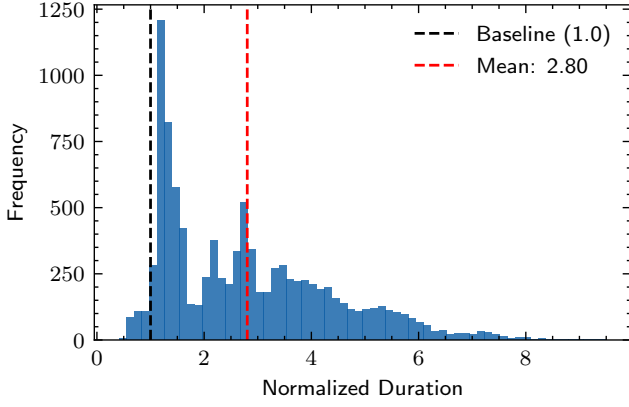
5

Fig. 3. Histogram of normalized optimal sentence determination times for GULPS over 10,000 random two-qubit unitaries using the instruction set $\mathcal{G} = \{CX, \sqrt[2]{CX}, \sqrt[3]{CX}\}$. Durations are normalized against the average runtime of Qiskit's `XXDecomposer`, which is marked by the black line at 1.0.
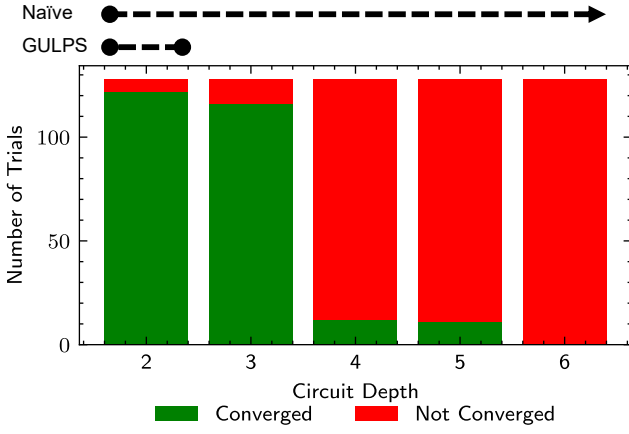

Fig. 4. Convergence rates for numeric synthesis across different circuit depths using the instruction set $\mathcal{G} = \{\sqrt[4]{iSWAP}\}$ when synthesizing a worst-case target at the apex of the corresponding circuit polytope. These results reflect the limitations of synthesis methods that attempt to optimize increasingly large parameterized circuits as a whole. In contrast, GULPS sidesteps this issue by repeatedly synthesizing guaranteed-feasible depth-2 segments, rendering convergence at higher depths irrelevant.

candidate sentences before a feasible solution is found. While GULPS is slightly slower on average, it provides richer information—namely, the intermediate invariants associated with each segment.

A major limitation of conventional numeric synthesis is that it must iterate over each candidate sentence and attempt to converge. If the optimizer fails, the sentence is rejected, and the process restarts from scratch with a different circuit structure. This makes sentence selection a bottleneck—particularly when the instruction set favors low-angle gates, as the time spent per rejection can be substantial. In practice, one must balance the risk of premature rejection against compilation time by increasing restarts or relaxing convergence tolerances.

To evaluate the effect of parameter dimensionality, we conducted a convergence experiment using the instruction set $\mathcal{G} = \{\sqrt[4]{iSWAP}\}$. We selected a worst-case target unitary located at the apex of the corresponding circuit polytope and attempted to synthesize it using global numeric synthesis at depths 2 through 6. Each trial used 128 optimizer restarts, a Levenberg–Marquardt solver with 2048 maximum iterations, and a residual threshold of 10^{-8} per term. This setup is comparable to existing numerical synthesis implementations, *e.g.* NuOp [15], though our implementation differs in both objective and solver method[1].

The results in Fig. 4 show that convergence success rates degrade rapidly with depth. While depths $\leqslant 3$ remain tractable, convergence fails almost entirely beyond depth 4. This presents a practical challenge for synthesis when the compiler must still attempt shallower depths first. Since convergence is already unreliable, it becomes difficult to determine whether a failed attempt reflects an invalid circuit or a failed optimization. *GULPS avoids this ambiguity by decomposing the synthesis into independently solved depth-2 segments with guaranteed feasibility.* Even though depth-2 synthesis may not converge on every attempt, we are assured that a solution exists, which allows us to invest more optimization effort, such as additional restarts or iterations, without wasting time on fundamentally invalid candidates.

Finally, several directions remain for extending this work. First, we believe it may be possible to eliminate the need for numerical synthesis entirely by developing fully analytical construction rules. For example, in the XX basis, inserting a single Z rotation between two XX gates is sufficient to control the invariant trajectory, as shown in [38]. This works because Z commutes with XX, allowing a single continuous parameter to steer the decomposition. Similar strategies may extend to the XX + YY basis, where inserted Y and X rotations can selectively modulate the XX and YY terms, respectively. However, generalizing this approach to the full XX + YY + ZZ basis—relevant for our interior segments with arbitrary starting points—is significantly more difficult. In this case, there is no unique conjugation axis, and interference between components complicates rule-based constructions. Recent work outlines a partial solution based on layered gate structures and synthesizing residual pieces to the canonical target invariants, but a general analytic synthesis rule remains an open problem [39]. We anticipate either analytical breakthroughs in these constructions or the development of differentiable surrogates using tools from parameterized quantum circuit optimization and quantum machine learning.

Second, GULPS could be extended to make more noise-aware or approximate synthesis choices. For example, prior work [38] has shown that decompositions can be routed to nearby invariant coordinates to reduce gate cost, especially when exact fidelity is not required. Additionally, if the monodromy LP yields multiple feasible solutions, it may

be possible to steer the decomposition toward lower-error sequences by incorporating fidelity-aware cost functions using a quadratic programming objective. Lastly, we aim to integrate GULPS into Qiskit by providing a full-featured implementation as a modular `Decomposer()` class, enabling ISA-aware synthesis pipelines within standard compiler workflows.

V. CONCLUSION

We introduced **GULPS** (Global Unitary Linear Programming Synthesis), a synthesis framework based on segmented Cartan trajectories, which breaks down the global two-qubit unitary synthesis problem into a series of depth-2 subproblems, each governed by linear constraints on canonical invariants. Our LP-based solver efficiently determines intermediate invariants, and a lightweight numerical routine recovers the interleaving local rotations. This hybrid approach guarantees feasibility for a wide range of gate sets, and avoids the convergence issues typical of conventional numerical synthesis.

ACKNOWLEDGMENTS

REFERENCES

[1] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum logic circuits," in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, 2005, pp. 272–275.

[2] B. Drury and P. Love, "Constructive quantum shannon decomposition from cartan involutions," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 39, p. 395305, 2008.

[3] M. Saeedi, M. Arabzadeh, M. S. Zamani, and M. Sedighi, "Block-based quantum-logic synthesis," *arXiv preprint arXiv:1011.2159*, 2010.

[4] D. Wierichs, M. West, R. T. Forestano, M. Cerezo, and N. Killoran, "Recursive cartan decompositions for unitary synthesis," *arXiv preprint arXiv:2503.19014*, 2025.

[5] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, vol. 6, no. 2, p. 021318, Jun. 2019. [Online]. Available: https://pubs.aip.org/apr/article/6/2/021318/570326/A-quantum-engineer-s-guide-to-superconducting

[6] K. X. Wei, I. Lauer, E. Pritchett, W. Shanks, D. C. McKay, and A. Javadi-Abhari, "Native two-qubit gates in fixed-coupling, fixed-frequency transmons beyond cross-resonance interaction," *PRX Quantum*, vol. 5, no. 2, p. 020338, 2024.

[7] D. R. Pérez, P. Varosy, Z. Li, T. Roy, E. Kapit, and D. Schuster, "Error-Divisible Two-Qubit Gates," *Physical Review Applied*, vol. 19, no. 2, p. 024043, Feb. 2023, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.19.024043

[8] D. G. Almeida, K. Ferris, N. Kanzawa, B. R. Johnson, and R. Davis, "Fractional gates reduce circuit depth at the utility scale | IBM Quantum Computing Blog," Nov. 2024. [Online]. Available: https://www.ibm.com/quantum/blog/fractional-gates

[9] B. Foxen, C. Neill, A. Dunsworth, P. Roushan, B. Chiaro, A. Megrant, J. Kelly, Z. Chen, K. Satzinger, R. Barends *et al.*, "Demonstrating a continuous set of two-qubit gates for near-term quantum algorithms," *Physical Review Letters*, vol. 125, no. 12, p. 120504, 2020.

[10] C. Scarato, K. Hanke, A. Remm, S. Lazăr, N. Lacroix, D. C. Zanuz, A. Flasby, A. Wallraff, and C. Hellings, "Realizing a continuous set of two-qubit gates parameterized by an idle time," *arXiv preprint arXiv:2503.11204*, 2025.

[11] M. Sugawara and T. Satoh, "Su (4) gate design via unitary process tomography: its application to cross-resonance based superconducting quantum devices," *arXiv preprint arXiv:2503.09343*, 2025.

[12] C. G. Yale, A. D. Burch, M. N. Chow, B. P. Ruzic, D. S. Lobser, B. K. McFarland, M. C. Revelle, and S. M. Clark, "Realization and calibration of continuously parameterized two-qubit gates on a trapped-ion quantum processor," *arXiv preprint arXiv:2504.06259*, 2025.

[13] F. Vatan and C. Williams, "Optimal Quantum Circuits for General Two-Qubit Gates," *Physical Review A*, vol. 69, no. 3, p. 032315, Mar. 2004, arXiv:quant-ph/0308006. [Online]. Available: http://arxiv.org/abs/quant-ph/0308006

[14] L. Madden and A. Simonetto, "Best approximate quantum compiling problems," *ACM Transactions on Quantum Computing*, vol. 3, no. 2, pp. 1–29, 2022, publisher: ACM New York, NY.

[15] L. Lao, P. Murali, M. Martonosi, and D. Browne, "Designing calibration and expressivity-efficient instruction sets for quantum computing," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2021, pp. 846–859.

[16] A. Javadi, "Improving quantum circuits with heterogenous gatesets," in *American Physical Society (March Meeting)*, 2023.

[17] E. McKinney, M. Hatridge, and A. K. Jones, "Mirage: Quantum circuit decomposition and routing collaborative design using mirror gates," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 704–718.

[18] J. Kalloor, M. Weiden, E. Younis, J. Kubiatowicz, B. De Jong, and C. Iancu, "Quantum hardware roofline: Evaluating the impact of gate expressivity on quantum processor design," in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2024, pp. 805–816.

[19] P. Rakyta and Z. Zimborás, "Approaching the theoretical limit in quantum gate decomposition," *Quantum*, vol. 6, p. 710, 2022.

[20] E. Younis, K. Sen, K. Yelick, and C. Iancu, "QFAST: Quantum Synthesis Using a Hierarchical Continuous Circuit Space," Mar. 2020, arXiv:2003.04462 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2003.04462

[21] E. Smith, M. G. Davis, J. M. Larson, E. Younis, L. Bassman, W. Lavrijsen, and C. Iancu, "LEAP: Scaling Numerical Optimization Based Synthesis Using an Incremental Approach," *ACM Transactions on Quantum Computing*, p. 3548693, Aug. 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3548693

[22] N. A. Nemkov, E. O. Kiktenko, I. A. Luchnikov, and A. K. Fedorov, "Efficient variational synthesis of quantum circuits with coherent multi-start optimization," *Quantum*, vol. 7, p. 993, May 2023, arXiv:2205.01121 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2205.01121

[23] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, "Geometric theory of nonlocal two-qubit operations," *Physical Review A*, vol. 67, no. 4, p. 042313, 2003, publisher: APS.

[24] R. R. Tucci, "An Introduction to Cartan's KAK Decomposition for QC Programmers," Jul. 2005, arXiv:quant-ph/0507171. [Online]. Available: http://arxiv.org/abs/quant-ph/0507171

[25] Y. Makhlin, "Nonlocal properties of two-qubit gates and mixed states, and the optimization of quantum computations," *Quantum Information Processing*, vol. 1, pp. 243–252, 2002.

[26] P. Watts, J. Vala, M. M. Müller, T. Calarco, K. B. Whaley, D. M. Reich, M. H. Goerz, and C. P. Koch, "Optimizing for an arbitrary perfect entangler: I. Functionals," *Physical Review A*, vol. 91, no. 6, p. 062306, Jun. 2015, arXiv: 1412.7347. [Online]. Available: http://arxiv.org/abs/1412.7347

[27] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, "Validating quantum computers using randomized model circuits," *Physical Review A*, vol. 100, no. 3, p. 032328, Sep. 2019, arXiv:1811.12926 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1811.12926

[28] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, "Minimum construction of two-qubit quantum operations," *Physical Review Letters*, vol. 93, no. 2, p. 020502, Jul. 2004, arXiv:quant-ph/0312193. [Online]. Available: http://arxiv.org/abs/quant-ph/0312193

[29] A. M. Childs, H. L. Haselgrove, and M. A. Nielsen, "Lower bounds on the complexity of simulating quantum gates," *Physical Review A*, vol. 68, no. 5, p. 052311, Nov. 2003. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.68.052311

[30] V. V. Shende, I. L. Markov, and S. S. Bullock, "Minimal universal two-qubit cnot-based circuits," *arXiv preprint quant-ph/0308033*, 2003.

[31] S. F. Lin, S. Sussman, C. Duckering, P. S. Mundada, J. M. Baker, R. S. Kumar, A. A. Houck, and F. T. Chong, "Let each quantum bit choose its basis gates," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2022, pp. 1042–1058.

[32] E. McKinney, C. Zhou, M. Xia, M. Hatridge, and A. K. Jones, "Parallel Driving for Fast Quantum Computing Under Speed Limits," in

Proceedings of the 50th Annual International Symposium on Computer Architecture, 2023, pp. 1–13.

[33] F. Preti, M. Schilling, S. Jerbi, L. M. Trenkwalder, H. P. Nautrup, F. Motzoi, and H. J. Briegel, "Hybrid discrete-continuous compilation of trapped-ion quantum circuits with deep reinforcement learning," *Quantum*, vol. 8, p. 1343, 2024.

[34] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient Z gates for quantum computing," *Physical Review A*, vol. 96, no. 2, Aug. 2017, publisher: American Physical Society (APS). [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.96.022330

[35] P. Watts, M. O'Connor, and J. Vala, "Metric structure of the space of two-qubit gates, perfect entanglers and quantum control," *Entropy*, vol. 15, no. 6, pp. 1963–1984, 2013.

[36] E. C. Peterson, G. E. Crooks, and R. S. Smith, "Fixed-depth two-qubit circuits and the monodromy polytope," *Quantum*, vol. 4, p. 247, 2020.

[37] A. A. Mele, "Introduction to haar measure tools in quantum information: A beginner's tutorial," *Quantum*, vol. 8, p. 1340, 2024.

[38] E. C. Peterson, L. S. Bishop, and A. Javadi-Abhari, "Optimal synthesis into fixed xx interactions," *Quantum*, vol. 6, p. 696, 2022.

[39] A. Wu, "Design the quantum instruction set with the cartan coordinate analysis framework," *arXiv preprint arXiv:2410.04008*, 2024.