Day 1: EchoKey *Cyclicity* → ZYZ Rewrite (Qiskit 1.4) — A Walkthrough

EchoKey Project

September 5, 2025

Abstract

This note documents the Day–1 pipeline that injects a symbolic single–qubit EchoKey operator, *Cyclicity*, into the Unitary Compiler Collection (UCC) and rewrites it to native $R_Z/R_Y/R_Z$ gates. We spell out the $su(2)$ algebra, the EchoKey frame, the forward map $\alpha = A^\top c$, the axis–angle interpretation, the ZYZ Euler synthesis, layout–aware per–wire indexing, and the verification protocol used to prove circuit–level equivalence (fidelity = 1 up to global phase).

1 Preliminaries: Pauli & $su(2)$

Let $\{\sigma_x, \sigma_y, \sigma_z\}$ be Pauli matrices and I_2 the identity. Useful identities:

$$\frac{1}{2}\operatorname{Tr}(\sigma_a\sigma_b) = \delta_{ab}, \qquad [\sigma_a, \sigma_b] = 2i\,\varepsilon_{abc}\,\sigma_c, \qquad \{\sigma_a, \sigma_b\} = 2\delta_{ab}I_2. \tag{1}$$

Every traceless Hermitian 2×2 matrix is of the form $H = \boldsymbol{\alpha}\cdot\boldsymbol{\sigma}$ with $\boldsymbol{\alpha} \in \mathbb{R}^3$ and $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$. The unitary exponential

$$U(\theta, \hat{\boldsymbol{n}}) \;=\; e^{-i\theta\,\hat{\boldsymbol{n}}\cdot\boldsymbol{\sigma}} \;=\; \cos\theta\, I_2 - i\sin\theta\,(\hat{\boldsymbol{n}}\cdot\boldsymbol{\sigma}) \tag{2}$$

effects a Bloch rotation by physical angle 2θ about axis $\hat{\boldsymbol{n}} \in \mathbb{S}^2$.

2 EchoKey local frame and forward map

We fix $K = 7$ EchoKey directions via an $A \in \mathbb{R}^{7\times 3}$ whose rows $A_{k\cdot} = \boldsymbol{a}_k^\top$ are unit vectors in $\mathbb{R}^3$. The traceless EchoKey basis operators are

$$E_k^\circ \;=\; \boldsymbol{a}_k \cdot \boldsymbol{\sigma} \quad (k = 1, \ldots, 7). \tag{3}$$

For a coefficient vector $\boldsymbol{c} \in \mathbb{R}^7$, the EchoKey Hamiltonian is

$$H_{\mathrm{EK}}(\boldsymbol{c}) \;=\; \sum_{k=1}^{7} c_k E_k^\circ \;=\; \Big(\sum_{k=1}^{7} c_k \boldsymbol{a}_k\Big)\cdot\boldsymbol{\sigma} \;=\; (A^\top \boldsymbol{c})\cdot\boldsymbol{\sigma}. \tag{4}$$

Right inverse and emergence of Pauli. If $\operatorname{rank}(A) = 3$, then $B = (A^\top A)^{-1}A^\top \in \mathbb{R}^{3\times 7}$ satisfies $BA = I_3$. Define $S_i := \sum_{k=1}^{7} B_{ik}E_k^\circ$ for $i \in \{x, y, z\}$. Then

$$S_i \;=\; \sum_k B_{ik}(\boldsymbol{a}_k\cdot\boldsymbol{\sigma}) \;=\; \Big(\sum_k B_{ik}\boldsymbol{a}_k\Big)\cdot\boldsymbol{\sigma} \;=\; (BA)_{i\cdot}\cdot\boldsymbol{\sigma} \;=\; \sigma_i. \tag{5}$$

Orthogonality $(1/2)\operatorname{Tr}(S_iS_j) = \delta_{ij}$ and commutators $[S_i, S_j] = 2i\varepsilon_{ijk}S_k$ follow. Thus the Pauli generators *emerge* from the EchoKey span.

3 Day–1 operator: *Cyclicity*

Pick $k = 1$ (row 0 in zero-based code) and set $\boldsymbol{a}_1 := A_{1\cdot}/\|A_{1\cdot}\|$. Define the symbolic gate

$$\text{ek\_cyc}(\theta) \;\equiv\; \exp\big(-i\,\theta\,E^{\circ}_{\text{cyc}}\big) = \exp\big(-i\,\theta\,\boldsymbol{a}_1\cdot\boldsymbol{\sigma}\big), \tag{6}$$

which implements a Bloch rotation of angle 2θ about axis $\hat{\boldsymbol{n}} = \boldsymbol{a}_1$.

4 Axis–angle to ZYZ Euler form

Any $U \in SU(2)$ admits a ZYZ decomposition $U = e^{-i\alpha\sigma_z/2}e^{-i\beta\sigma_y/2}e^{-i\gamma\sigma_z/2}$. For Day–1 we instantiate the exact U from (2) with $\hat{\boldsymbol{n}} = \boldsymbol{a}_1$ and θ, and synthesize the ZYZ gate sequence using Qiskit's stable `OneQubitEulerDecomposer("ZYZ")`. *In the code we do not hand–derive* (α, β, γ)*; we ask the decomposer to synthesize the* ***circuit*** *for* U *exactly, avoiding angle–branch pitfalls.*

5 Layout awareness (logical → physical)

Compilation can permute qubits. Let π map a virtual/logical qubit q to a physical wire $\pi(q)$ (`final_layout` in Qiskit 1.4). Since $\boldsymbol{a}_1$ is a *per–wire* attribute, we must fetch $A[\pi(q)]$ before computing the axis for that gate instance. The pass reads π from `self.property_set` and uses $\boldsymbol{a}_1(\pi(q))$ in synthesis.

6 Rewrite rule (as implemented)

For each node `ek_cyc(`θ`)` on qubit q:

1. Resolve $p = \pi(q)$ if a layout exists; else $p = q$.
2. Load $A^{(p)}$ for wire p and set $\hat{\boldsymbol{n}} = \boldsymbol{a}_1^{(p)}/\left\|\boldsymbol{a}_1^{(p)}\right\|$.
3. Form $U = \cos\theta\, I_2 - i\sin\theta\,(\hat{\boldsymbol{n}}\cdot\boldsymbol{\sigma})$.
4. Use the ZYZ decomposer to synthesize a circuit D over $\{R_Z, R_Y\}$ such that $D \sim U$ (up to phase).
5. Substitute the node with D (i.e., $R_Z(\alpha); R_Y(\beta); R_Z(\gamma)$).

7 Verification protocol

Because Qiskit cannot natively evaluate our symbolic gate, we build a *materialized reference* circuit by replacing ek_cyc(θ) with the exact UnitaryGate(U) from step 3 above. We then run only our rewrite pass (no layout/routing) and compare unitaries using

$$\mathcal{F}(U, V) \;=\; \frac{1}{2^n}\left|\text{Tr}\Big(U^{\dagger}V\Big)\right|, \tag{7}$$

which is insensitive to global phase. All Day–1 examples (1–4 qubits; random batteries) report $\mathcal{F} \approx 1$, certifying equivalence of the ZYZ rewrite to the exact unitary specification.

8 Practical implications (Day–1)

- **Basis–native output:** emitting $R_Z/R_Y/R_Z$ preserves compiler optimizations (cancellation, commutation, hardware targeting) that opaque unitaries block.
- **Per–wire semantics:** axes can depend on the physical site, enabling alignment with device frames.
- **Clean injection point:** authors write in EchoKey vocabulary; the pass maps to Pauli–native gates.
- **Test harness pattern:** *materialize–then–compare* generalizes to future operators.

9 Toward Days 2–8

Day–2 introduces a second generator (row 1), Day–3 a third, etc. With $B = (A^\top A)^{-1} A^\top$ we define $S_i = \sum_k B_{ik} E_k^\circ$ and verify orthonormality and $su(2)$ commutators. The Day–8 note shows how Pauli operators *emerge* from the EchoKey family (span=3, right inverse exists, commutation closes).

Appendix A: Quick algebra checks

Orthogonality. With $S_i = \sum_k B_{ik} E_k^\circ$ and $BA = I_3$,

$$\frac{1}{2}\,\mathrm{Tr}(S_i S_j) = \frac{1}{2}\sum_{k,\ell} B_{ik} B_{j\ell}\,\mathrm{Tr}\left((\boldsymbol{a}_k \cdot \boldsymbol{\sigma})(\boldsymbol{a}_\ell \cdot \boldsymbol{\sigma})\right) \tag{8}$$

$$= \sum_{k,\ell} B_{ik} B_{j\ell}\, \boldsymbol{a}_k \cdot \boldsymbol{a}_\ell \;=\; (BAA^\top B^\top)_{ij} \;=\; \delta_{ij}. \tag{9}$$

Commutators. Using $[\boldsymbol{u}\cdot\boldsymbol{\sigma}, \boldsymbol{v}\cdot\boldsymbol{\sigma}] = 2i(\boldsymbol{u}\times\boldsymbol{v})\cdot\boldsymbol{\sigma}$,

$$[S_i, S_j] \;=\; 2i\Big(\sum_k B_{ik}\boldsymbol{a}_k \times \sum_\ell B_{j\ell}\boldsymbol{a}_\ell\Big)\cdot\boldsymbol{\sigma} \;=\; 2i\big((BA)_{i\cdot} \times (BA)_{j\cdot}\big)\cdot\boldsymbol{\sigma} \;=\; 2i\,\varepsilon_{ijk} S_k. \tag{10}$$

Appendix B: Implementation sketch (Qiskit 1.4)

1. Define `EchoKeyCyclicityGate(theta)` with an empty body.
2. Prepare per–wire weights `weights = {phys_q: SiteWeights(A)}`.
3. In the pass: get `layout = self.property_set.get("final_layout") or ...`, then map virtual to physical.
4. Build U as in (2) with $\hat{\boldsymbol{n}} = \boldsymbol{a}_1$ and θ.
5. Call `OneQubitEulerDecomposer("ZYZ")(U)` to obtain a 1–qubit circuit; substitute the node.
6. For tests: materialize the original circuit by inserting `UnitaryGate(U)` in place of the symbolic gate, then compare unitaries.