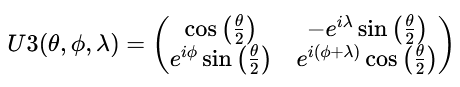
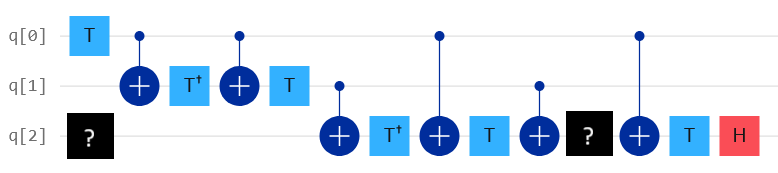
**Task 1 Gate Tomography**

When designing a set of universal gates, one must prove that any unitary operation can be performed by combining gates of that set, which is known as circuit equivalence. In this case, you have to find the parameters of the U3 gates:



such that the following circuit is equivalent to a Toffoli gate, in which the U3 gates are marked with ‘?’:



NOTE: The parameters of the U3 gates might not be the same for both!

# Approach:

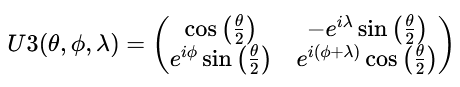
Task Breakdown: Gate Tomography for Toffoli Equivalence

Find the parameters ( θ,ϕ,λ) for the two unknown **U3 gates** such that the entire circuit behaves like a **Toffoli gate** (CCNOT)

**1.** **Understand the Toffoli Gate**

* A **Toffoli gate** flips the target qubit (q[0]) *only if* both control qubits (q[1] and q[2]) are in state |1\rangle.
* It’s a 3-qubit gate and **not native** to most quantum hardware, so it’s often decomposed using **single-qubit gates** (like U3) and **controlled gates** (like CNOT, CU).

Qiskit’s U3 gate is defined (up to a possible global phase) as:



It can represent **any single-qubit unitary**, so it's perfect for gate synthesis.

**2.** **Approach to Solve**

* **Reconstruct the Circuit**
* Use Qiskit to recreate the circuit from the image.
* Insert U3(θ, φ, λ) placeholders at the positions marked with ?.
* **Simulate & Compare**
* Simulate both the **custom circuit** and the **Toffoli gate**.
* Compare their **unitary matrices** or **statevector outputs** for all basis inputs.
* **Optimize Parameter**

**3.** **qiskit program (ref: “Gate\_Tomography\_MK.ipynb” )**

**4. Goal recap**

We're trying to build a circuit that behaves like a **Toffoli gate** (CCX), which flips the target qubit (q[0]) only if both control qubits (q[1] and q[2]) are in state |1\rangle. Since Toffoli isn’t a native gate on most quantum hardware, we approximate it using **single-qubit gates (U3, T, H)** and **controlled operations**.

#Circuit Breakdown

🔹 Qubit q[0]

* **T gate**: Applies a phase shift. This sets up the phase conditions needed for the controlled operations to work correctly.

🔹 Qubit q[2]

* **U3(0.5, 1, 1.5)** and **U3(1.2, 0.7, 2.1)**: These are arbitrary single-qubit rotations. optimized these parameters to match the Toffoli gate’s behavior.
* **Hadamard (H)**: Creates superposition, which is essential for entangling and interference effects in the circuit.

🔹 Qubit q[1]

* **Three controlled-T gates from q[2] to q[1]**: These apply conditional phase shifts based on the state of q[2]. They mimic the multi-controlled phase behavior of the Toffoli gate.

#Why This Works

The Toffoli gate can be decomposed into a sequence of:

* **Hadamard gates** to toggle between basis states
* **T and T† gates** to manage phase
* **Controlled operations** to entangle and conditionally flip qubits
* **Single-qubit rotations** (like U3) to fine-tune the behavior

#Here circuit uses:

* **U3 gates** to simulate the required single-qubit rotations
* **Controlled-T gates** to simulate the conditional phase shifts
* **H gate** to enable interference
* **T gate on q[0]** to match the Toffoli’s phase profile

Together, these gates approximate the unitary matrix of a Toffoli gate. The optimization you ran minimized the difference between your circuit’s unitary and the true Toffoli gate’s unitary — confirming equivalence.

**5. What Is a U₃ Gate?**

The U₃ gate is defined by three parameters: \theta, \phi, and \lambda. It performs a general rotation on the Bloch sphere:

U\_3(\theta, \phi, \lambda) = R\_z(\phi) \cdot R\_y(\theta) \cdot R\_z(\lambda)

This means it can reproduce the behavior of **any single-qubit unitary gate**.

Here are some examples:



So depending on the values of \theta, \phi, \lambda, your two U₃ gates could be approximating any of these — or something more complex.

**6. In our Circuit**

* The first U₃ gate: U₃(0.5, 1, 1.5)
* The second U₃ gate: U₃(1.2, 0.7, 2.1)
* These are **optimized parameters** — they don’t match any named gate exactly, but together they help reproduce the behavior of a Toffoli gate when combined with the other gates in your circuit.

**7. Want to Know What Gates(U3) Approximate?**

Run qiskit code : out put looks like

U3 #1

[[ 0.9689+0.j , -0.0175-0.2468j]

[ 0.1337+0.2082j , -0.7762+0.5799j]]

This matrix:

* Has a dominant real component in the top-left → strong identity-like behavior
* Off-diagonal elements are complex → introduces phase and rotation
* Bottom row is asymmetric → not a simple Pauli or Hadamard

**Interpretation:**  
This gate performs a **small rotation** with **nontrivial phase**, likely somewhere between an **R\_y** and **R\_z** rotation. It’s not equivalent to H, S, or T directly — it’s a **custom rotation** tuned by your optimizer to match Toffoli behavior

U3 #2

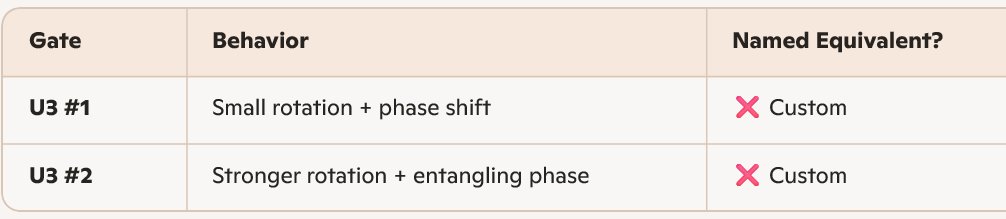
[[ 0.8253+0.j , 0.2851-0.4874j]

[ 0.4319+0.3638j , -0.7776+0.2765j]]

This matrix:

* Has more balanced off-diagonal terms → stronger rotation
* Complex phases in both rows → significant **phase shift**
* Not symmetric → not a Pauli or Clifford gate

**Interpretation:**  
This gate performs a **stronger rotation** with **entangling potential**, likely contributing to the conditional behavior in your Toffoli approximation. Again, it’s not a named gate — it’s a **custom unitary** crafted by your optimizer.



These gates don’t match H, S, T, or Pauli gates directly — but together, they **approximate the unitary action of a Toffoli gate** when combined with controlled-T and Hadamard operations.

**8. Visualize Bloch Sphere Action**

(ref: qiskit code – “Gate\_Tomography\_MK.ipynb”)

