Quantum Circuit Synthesis and Compiler

Yupan Liu
Shuxiang Cao
Supervisor: Junde Wu

December 4, 2015

Quantum circuit synthesis

Superconducting quantum computing and DQC1

Implementation of quantum algorithms

Quantum state and quantum gate

- Quantum state: single-qubit

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis.

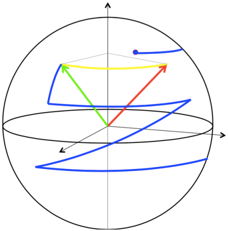
Quantum state and quantum gate

- Quantum state: single-qubit

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis.

- Bloch sphere representation

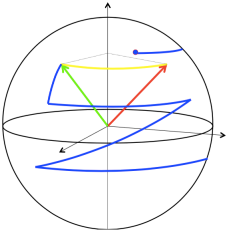


Quantum state and quantum gate

- Quantum state: single-qubit

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

  where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis.
- Bloch sphere representation



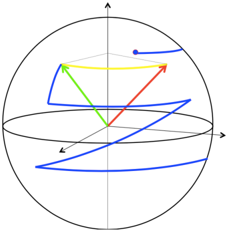
- Quantum gate: unitary

Quantum state and quantum gate

- Quantum state: single-qubit

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

  where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis.
- Bloch sphere representation



- Quantum gate: unitary
- n-qubit case: tensor product

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea?

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea? Using **algebraic** number theory, $O(log(\frac{1}{\epsilon}))$!

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea? Using **algebraic** number theory, $O(log(\frac{1}{\epsilon}))$!
- Decomposition: $SU(2)$

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea? Using **algebraic** number theory, $O(log(\frac{1}{\epsilon}))$!
- Decomposition: $SU(2) \Rightarrow$ Euler angle

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea? Using **algebraic** number theory, $O(log(\frac{1}{\epsilon}))$!
- Decomposition: $SU(2) \Rightarrow$ Euler angle $\Rightarrow$ z-rotation

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea? Using **algebraic** number theory, $O(log(\frac{1}{\epsilon}))$!
- Decomposition: $SU(2) \Rightarrow$ Euler angle $\Rightarrow$ z-rotation
- Approximation of z-rotation $\Rightarrow$ Grid problem

$$R_z(\theta) = e^{-i\theta Z/2} = \begin{pmatrix} e^{-i\theta/2} & 0 & \\ 0 & e^{i\theta/2} & 0 \end{pmatrix}$$

$$U = \frac{1}{\sqrt{2}^k}\begin{pmatrix} u & -t^\dagger \\ t & u^\dagger \end{pmatrix}, u, t \in \mathbb{Z}[i, \frac{1}{\sqrt{2}}]$$

$$\|R_z(\theta) - U\| < \epsilon$$

Approximation of single-qubit gate

- Solovay-Kitaev algorithm based on Lie group, $O(log^c(\frac{1}{\epsilon}))$
- Better idea? Using **algebraic** number theory, $O(log(\frac{1}{\epsilon}))$!
- Decomposition: $SU(2) \Rightarrow$ Euler angle $\Rightarrow$ z-rotation
- Approximation of z-rotation $\Rightarrow$ Grid problem

$$R_z(\theta) = e^{-i\theta Z/2} = \begin{pmatrix} e^{-i\theta/2} & 0 & \\ 0 & e^{i\theta/2} & 0 \end{pmatrix}$$

$$U = \frac{1}{\sqrt{2}^k} \begin{pmatrix} u & -t^\dagger \\ t & u^\dagger \end{pmatrix}, u, t \in \mathbb{Z}[i, \frac{1}{\sqrt{2}}]$$

$$\|R_z(\theta) - U\| < \epsilon$$

- Implementation using python and sympy.
  Ref Quantum Information & Computation, 2015, 15(1-2): 159-180.

Exact synthesis of Clifford+T single-qubit gate

- Clifford+T gate $\Rightarrow$ Complete basis

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}, \omega = e^{i\pi/4}$$

Exact synthesis of Clifford+T single-qubit gate

- Clifford+T gate $\Rightarrow$ Complete basis

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}, \omega = e^{i\pi/4}$$

- Single-qubit Clifford+T gate $\Leftrightarrow$ Unitaries on $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$

Exact synthesis of Clifford+T single-qubit gate

- Clifford+T gate $\Rightarrow$ Complete basis

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}, \omega = e^{i\pi/4}$$

- Single-qubit Clifford+T gate $\Leftrightarrow$ Unitaries on $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$
- Decomposition of Clifford+T gate

$$(T|\varepsilon)(HT|SHT)^*C$$

Exact synthesis of Clifford+T single-qubit gate

- Clifford+T gate $\Rightarrow$ Complete basis

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}, \omega = e^{i\pi/4}$$

- Single-qubit Clifford+T gate $\Leftrightarrow$ Unitaries on $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$
- Decomposition of Clifford+T gate

$$(T|\varepsilon)(HT|SHT)^{*}C$$

- Implementation using python and sympy.
Ref arXiv preprint arXiv:1312.6584, 2013.

Multi-qubit gate synthesis

- n-qubit gate: $2^n \times 2^n$ unitary

Multi-qubit gate synthesis

- n-qubit gate: $2^n \times 2^n$ unitary
- n-qubit gate $\Rightarrow \frac{N(N-1)}{2}$ two-level unitary, $N = 2^n$

Multi-qubit gate synthesis

- n-qubit gate: $2^n \times 2^n$ unitary
- n-qubit gate $\Rightarrow \frac{N(N-1)}{2}$ two-level unitary, $N = 2^n$
- two-level unitary $\Rightarrow$ CNOT+single-qubit gate, $O(n^2)$

Multi-qubit gate synthesis

- n-qubit gate: $2^n \times 2^n$ unitary
- n-qubit gate $\Rightarrow \frac{N(N-1)}{2}$ two-level unitary, $N = 2^n$
- two-level unitary $\Rightarrow$ CNOT+single-qubit gate, $O(n^2)$
- Implementation using python and sympy.
  Ref Physical Review A, 1995, 52(5): 3457.

Quantum circuit synthesis

- Quantum state and quantum gate
- Single-qubit gate synthesis
- Multi-qubit gate synthesis
- Next?

Superconducting quantum computing and DQC1

Implementation of quantum algorithms

Next?

- Clifford+T single-qubit gate synthesis: too slow

Next?

- Clifford+T single-qubit gate synthesis: too slow

  Because of sympy.

Next?

- Clifford+T single-qubit gate synthesis: too slow
  Because of sympy.
- Efficient exact decomposition on multiqubit gate

Next?

- Clifford+T single-qubit gate synthesis: too slow
  Because of sympy.

- Efficient exact decomposition on multiqubit gate

Multi-qubit case

Multi-qubit Clifford+T gate $\Leftrightarrow$ Unitaries on $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$

Ref Physical Review A, 2013, 87(3): 032332.

Next?

- Clifford+T single-qubit gate synthesis: too slow

  Because of sympy.

- Efficient exact decomposition on multiqubit gate

Multi-qubit case

Multi-qubit Clifford+T gate $\Leftrightarrow$ Unitaries on $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$

Ref Physical Review A, 2013, 87(3): 032332.

More efficient approach?

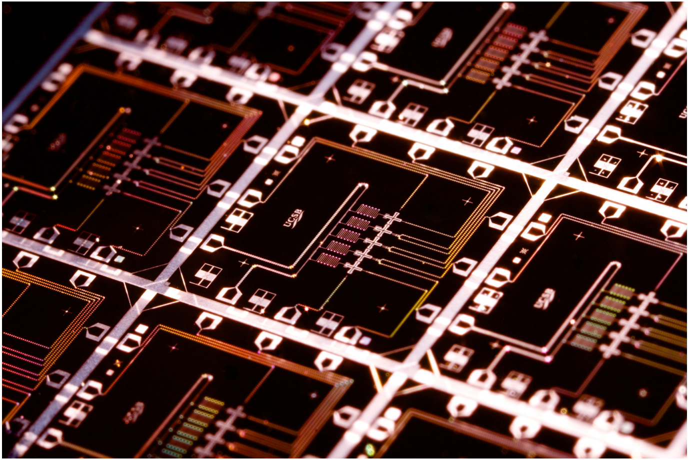
Quantum circuit synthesis

Superconducting quantum computing and DQC1

Implementation of quantum algorithms

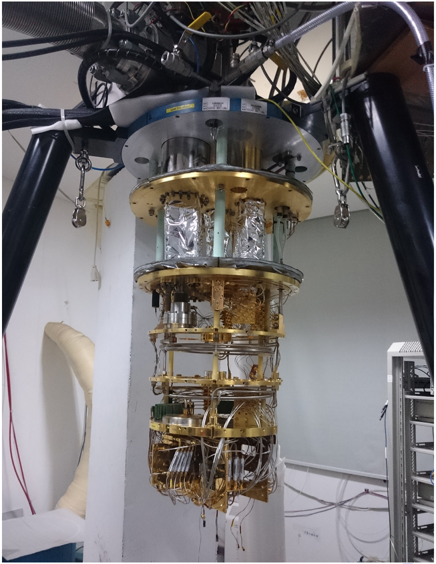
Introduction to superconducting qubit

- XMon Qubit (UCSB)



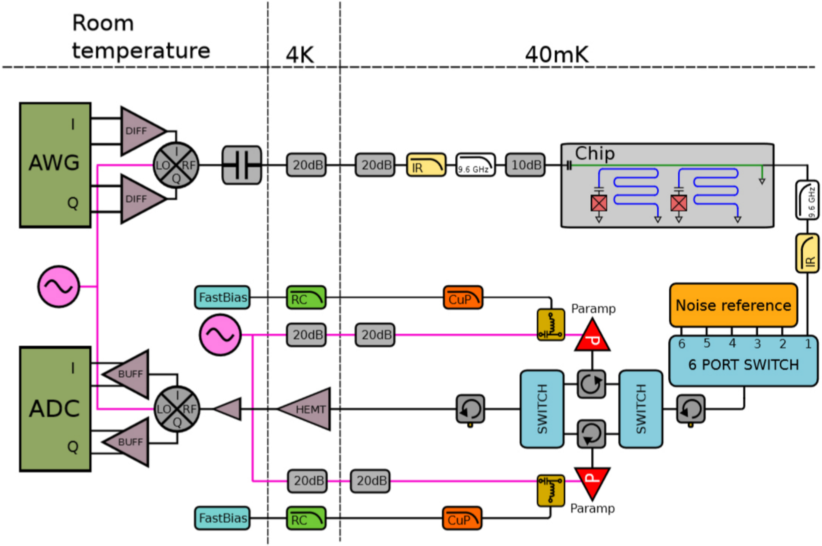
Introduction to superconducting qubit(Cont.)

- Refrigerator (ZJU SQCG Group)



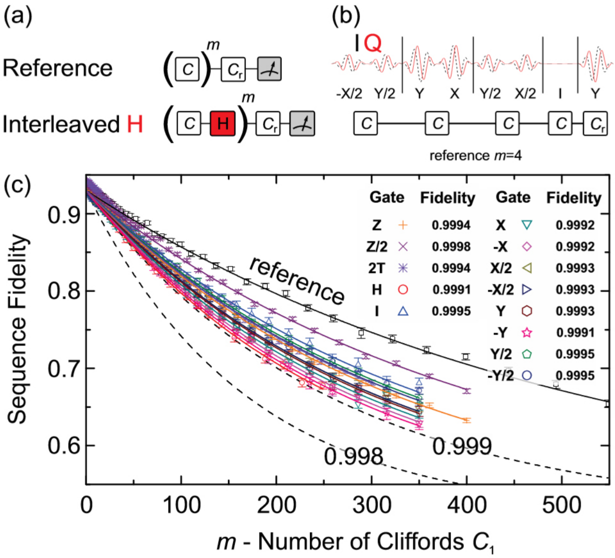
Introduction to superconducting qubit(Cont.)

- Measure System (UCSB)



Introduction to superconducting qubit(Cont.)

- Clifford Gate Benchmark (UCSB)



Quantum circuit synthesis

Superconducting quantum computing and DQC1

- Introduction to superconducting qubit
- Superconducting quantum computing and DQC1
- Next?

Implementation of quantum algorithms

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)
  - 6 physical qubits $\Rightarrow$ Error-correction code, 2 logical qubits

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)
  - 6 physical qubits $\Rightarrow$ Error-correction code, 2 logical qubits
  - Decoherence time $10\mu s$ $\Rightarrow$ #Quantum gate$\approx 10^3$

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)
  - 6 physical qubits $\Rightarrow$ Error-correction code, 2 logical qubits
  - Decoherence time $10\mu s \Rightarrow$ #Quantum gate$\approx 10^3$

Ref arXiv preprint arXiv:1410.5793, 2014.

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)
  - 6 physical qubits $\Rightarrow$ Error-correction code, 2 logical qubits
  - Decoherence time $10\mu s$ $\Rightarrow$ #Quantum gate$\approx 10^3$

Ref arXiv preprint arXiv:1410.5793, 2014.

- DQC1(Deterministic Quantum Computation with 1-qubit)
  - A computational complexity with one-qubit and an ancilla

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)
  - 6 physical qubits $\Rightarrow$ Error-correction code, 2 logical qubits
  - Decoherence time $10\mu s$ $\Rightarrow$ #Quantum gate$\approx 10^3$

Ref arXiv preprint arXiv:1410.5793, 2014.

- DQC1(Deterministic Quantum Computation with 1-qubit)
  - A computational complexity with one-qubit and an ancilla
  - DQC1-Complete Problem:
    - Quantum Fourier transform
    - Approximation of Jones Polynomial

Superconducting quantum computing and DQC1

- Superconducting qubit XMon, based on LabRad(UCSB)
  - 6 physical qubits $\Rightarrow$ Error-correction code, 2 logical qubits
  - Decoherence time $10\mu s \Rightarrow \#$Quantum gate$\approx 10^3$

Ref arXiv preprint arXiv:1410.5793, 2014.

- DQC1(Deterministic Quantum Computation with 1-qubit)
  - A computational complexity with one-qubit and an ancilla
  - DQC1-Complete Problem:
    - Quantum Fourier transform
    - Approximation of Jones Polynomial

Ref Physical review letters, 2000, 85(14): 3049.

Next?

- Optimization of quantum circuit synthesis on DQC1

Next?

- Optimization of quantum circuit synthesis on DQC1
- Practical quantum computer
  - using QubitServer on LabRad
  - using quantum circuit synthesis

Next?

- Optimization of quantum circuit synthesis on DQC1
- Practical quantum computer
  - using QubitServer on LabRad
  - using quantum circuit synthesis
- Publication?

Quantum circuit synthesis

Superconducting quantum computing and DQC1

Implementation of quantum algorithms

Quantum circuit synthesis

Superconducting quantum computing and DQC1

Implementation of quantum algorithms

- Example: Quantum Fourier transform
- Implementation of quantum Fourier transform
- Other algorithms and applications

Introduction to quantum Fourier transform

- Discrete Fourier Transform

$$\tilde{f}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k/N} f_j$$

Introduction to quantum Fourier transform

- Discrete Fourier Transform

$$\tilde{f}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k / N} f_j$$

- Quantum Fourier Transform

$$|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k / N} |j\rangle$$

Introduction to quantum Fourier transform

- Discrete Fourier Transform

$$\tilde{f}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jk/N} f_j$$

- Quantum Fourier Transform

$$|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jk/N} |j\rangle$$

$$|\tilde{\phi}\rangle = \hat{F}|\phi\rangle, \hat{F}^{\dagger}\hat{F} = \hat{I}$$

Introduction to quantum Fourier transform

- Discrete Fourier Transform

$$\tilde{f}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j$$

- Quantum Fourier Transform

$$|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} |j\rangle$$

$$|\tilde{\phi}\rangle = \hat{F}|\phi\rangle, \hat{F}^\dagger \hat{F} = \hat{I}$$

$$\hat{F} = \sum_{j,k=0}^{N-1} \frac{e^{2\pi ijk/N}}{\sqrt{N}} |k\rangle\langle j|$$

Quantum circuit synthesis

Superconducting quantum computing and DQC1

Implementation of quantum algorithms

- Example: Quantum Fourier transform
- Implementation of quantum Fourier transform
- Other algorithms and applications

Quantum Fourier transform and quantum circuit synthesis

- Consider 3-qubit case:

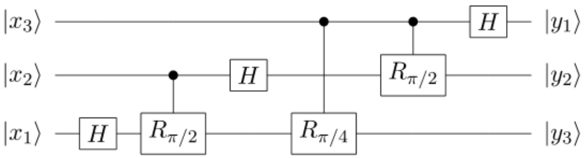
$$|x_1, x_2, x_3\rangle = \frac{1}{\sqrt{2^3}}(|0\rangle + e^{2\pi i[0.x_3]}|1\rangle) \otimes (|0\rangle + e^{2\pi i[0.x_2x_3]}|1\rangle) \\ \otimes (|0\rangle + e^{2\pi i[0.x_1x_2x_3]}|1\rangle)$$

Quantum Fourier transform and quantum circuit synthesis

- Consider 3-qubit case:

$$|x_1, x_2, x_3\rangle = \frac{1}{\sqrt{2^3}}(|0\rangle + e^{2\pi i[0.x_3]}|1\rangle) \otimes (|0\rangle + e^{2\pi i[0.x_2x_3]}|1\rangle) \otimes (|0\rangle + e^{2\pi i[0.x_1x_2x_3]}|1\rangle)$$

- Quantum circuit implementation of 3-qubit QFT

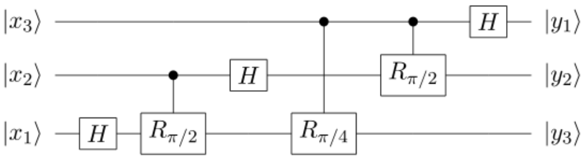


Quantum Fourier transform and quantum circuit synthesis

- Consider 3-qubit case:

$$|x_1, x_2, x_3\rangle = \frac{1}{\sqrt{2^3}}(|0\rangle + e^{2\pi i[0.x_3]}|1\rangle) \otimes (|0\rangle + e^{2\pi i[0.x_2x_3]}|1\rangle) \\ \otimes (|0\rangle + e^{2\pi i[0.x_1x_2x_3]}|1\rangle)$$

- Quantum circuit implementation of 3-qubit QFT



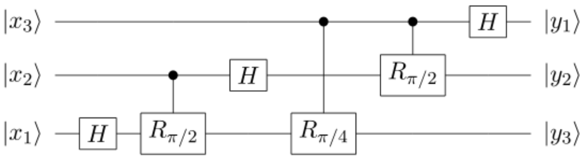
- Compiler?

Quantum Fourier transform and quantum circuit synthesis

- Consider 3-qubit case:

$$|x_1, x_2, x_3\rangle = \frac{1}{\sqrt{2^3}}(|0\rangle + e^{2\pi i[0.x_3]}|1\rangle) \otimes (|0\rangle + e^{2\pi i[0.x_2x_3]}|1\rangle)$$
$$\otimes (|0\rangle + e^{2\pi i[0.x_1x_2x_3]}|1\rangle)$$

- Quantum circuit implementation of 3-qubit QFT



- Compiler?
  Implement quantum algorithm by quantum circuit synthesis!

Other algorithms and applications

Quantum simulation

Using Jordan-Wigner transform to simulation Fermion quantum system.

Ref Physical Review A, 2001, 64(2): 022319.

Other algorithms and applications

Quantum simulation

Using Jordan-Wigner transform to simulation Fermion quantum system.

Ref Physical Review A, 2001, 64(2): 022319.

Backtracking algorithm

Using quantum random walk to speed up backtracking algorithm.

Ref arXiv preprint arXiv:1509.02374, 2015.

Other algorithms and applications

Quantum simulation

Using Jordan-Wigner transform to simulation Fermion quantum system.

Ref Physical Review A, 2001, 64(2): 022319.

Backtracking algorithm

Using quantum random walk to speed up backtracking algorithm.

Ref arXiv preprint arXiv:1509.02374, 2015.

Simulate open quantum system

Consider quantum channel(superoperator)'s representation, using quantum circuit synthesis to decompose single-qubit quantum channels.

Ref Physical review letters, 2013, 111(13): 130504.

Thanks for listening!

Q & A