5 量子回路の自動設計手法

中島 裕美〔日本電信電話（株）NTT コミュニケーション科学基礎研究所〕
河野 泰人〔日本電信電話（株）NTT コミュニケーション科学基礎研究所〕
関川　浩〔日本電信電話（株）NTT コミュニケーション科学基礎研究所〕

現在のコンピュータの演算は，AND, OR, NOT などの基本演算を組み合わせることで実現できる．同様に，量子コンピュータでは，1 量子ビットの演算と 2 量子ビットの演算を基本演算とし，これらを組み合わせることで，所望の n 量子ビットの演算を実現できる．いま，量子コンピュータの基本演算を，現在のコンピュータの論理回路のように記号を用いて表すことにすると，与えられた演算を実現する基本演算の構成を示した図が量子回路である．

入出力の量子ビットの状態をベクトルで表すと，量子コンピュータの演算は行列で表現することができる．n 量子ビットの演算に対応する行列の大きさは $2^n \times 2^n$ である．基本演算は，2×2（1 量子ビットの演算），または 4×4（2 量子ビットの演算）の行列で表現できる．したがって，n 量子ビットの演算は，基本演算に対応する行列の積で書くことができる．本稿では，この行列表現を用いて，与えられた n 量子ビットの演算（$2^n \times 2^n$ 行列）から量子回路を自動設計する手法を紹介する．量子回路は，量子コンピュータ上の実行命令の系列に対応しているので，本手法は，言い換えれば，量子コンピュータの演算を実行命令の系列に自動翻訳する手法である．利用する技術は行列の特異値分解である．

効率的な演算と量子回路

量子コンピュータは，因数分解を高速に解くアルゴリズム（Shor のアルゴリズム）が示されたことで，現在のコンピュータよりも高い計算能力を持つコンピュータとして注目されるようになった．では，量子コンピュータを使うと，すべての演算が現在のコンピュータよりも格段に速くなるのだろうか．実際にはそうではなく，量子コンピュータによって高速になる演算と，現在のコンピュータとあまり変わらない演算がある．

量子コンピュータの計算時間は，大まかには，与えられた演算を実行するために必要な基本演算の数に対応している．量子コンピュータにおける基本演算は，1 量子ビットの演算と 2 量子ビットの演算である．これらの基本演算を組み合わせることで，任意の n 量子ビットの演算を作ることができる．一般には，任意の n 量子ビットの演算を構成するためには，$O(4^n)$ の基本演算が必要である [2]．量子ビット数 n に対して，指数関数的に計算時間が増大するので，効率的な演算とはいえない．これに対して，Shor のアルゴリズムで用いられている量子 Fourier 変換は，量子ビット数 n に対して，必要な基本演算数が $O(n^2)$ であるため，量子コンピュータ上で効率的に実行することができる．

量子回路とは，図 -1 のように，与えられた演算を実現する基本演算の構成を記号で表した図である．基本演算は，図 -2 に示す記号で記述される．横に通った各々の線が 1 量子ビットに対応しており，線上の記号がその量子ビットに適用される基本演算で，左から時系列に並べられている．量子回路に従って，左から順に基本演算を行うと，所望の出力状態まで到達できる．したがって，量子回路は実行命令の系列を表しているともいえる．

本稿では，n 量子ビットの演算から，量子回路を自動設計する手法について解説する．ここでは，量子コンピュータの演算を行列で表現したものを利用する．そこでまず，次章では，量子コンピュータの演算と行列計算の関係について説明する．次に，行列の特異値分解を応用

した量子回路の自動設計手法を紹介し，例として，量子Fourier変換の量子回路の設計を挙げる．最後に今後の課題について述べる．

量子コンピュータの演算と行列計算

量子コンピュータにおける演算とは，与えられた量子ビット状態を，目的とする量子ビットの状態へ遷移させるプロセスである．数学的には，量子ビットの量子状態をベクトル，演算を行列で書くことができる．たとえば1量子ビットの重ね合わせ状態を $\alpha|0\rangle + \beta|1\rangle$ とすると，これは以下の2次元ベクトルで書ける．

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}.$$

ここで，量子状態を表すベクトルは，長さが1（$|\alpha|^2 + |\beta|^2 = 1$，$\alpha$，$\beta$ は複素数）である．n 量子ビットの量子状態は，テンソル積を用いて書ける．たとえば2量子ビットで，第1量子ビットの量子状態が $\alpha_1|0\rangle + \beta_1|1\rangle$，第2量子ビットの量子状態が $\alpha_2|0\rangle + \beta_2|1\rangle$ の場合は，以下のような4次元ベクトルで表せる．

$$\begin{pmatrix}\alpha_1\\ \beta_1\end{pmatrix}\otimes\begin{pmatrix}\alpha_2\\ \beta_2\end{pmatrix} = \begin{pmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{pmatrix}. \qquad (1)$$

ベクトルの次元は，n 量子ビットのとき 2^n 次元になる．

量子計算は，与えられた量子状態（ベクトル）から別の量子状態（ベクトル）への写像を表す行列として記述できる．ここで，入出力のベクトルはどちらも長さが1であるため，演算を表す行列は長さを変えない線形作用素，すなわちユニタリ行列となる．したがって，行列 U の共役転置行列（$U^\dagger$）は，逆行列（U^{-1}）と一致する．

たとえば，入力状態を $\alpha|0\rangle + \beta|1\rangle$ とすると，図-2に示すNOTの演算は，以下のように計算できる．

$$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\beta\\ \alpha\end{pmatrix}.$$

これは，与えられた量子状態 $\alpha|0\rangle + \beta|1\rangle$ を $\beta|0\rangle + \alpha|1\rangle$ という0と1に対応する確率振幅（ベクトルの要素）が反転した状態へ遷移させる演算であるため，NOTに対応していることが分かる．

次に，2量子ビットの演算を考える．入力状態を（1）とし，第1量子ビットを制御ビット，第2量子ビットを標的ビットとする制御NOT演算を考える．

$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{pmatrix} = \begin{pmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\beta_2\\ \beta_1\alpha_2\end{pmatrix},$$

より，制御NOTは，第1量子ビットが1のときの第2量子ビットの状態（$|10\rangle$ と $|11\rangle$ に対応するベクトルの値）を反転させる演算である．論理回路に対応させると，ちょうど第2量子ビットの出力が第1量子ビットと第2量子ビットのXORになる．量子回路の場合，入力と出力の線の数（量子ビット数）は同じであるため，XORのような2入力1出力の演算を実現するために，制御NOTのような2入力2出力の演算を用いる．

一般に，n 量子ビットに対する量子計算は，2^n 次元ベ

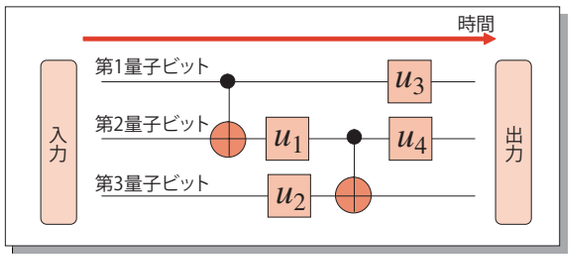

図-1　量子回路の例

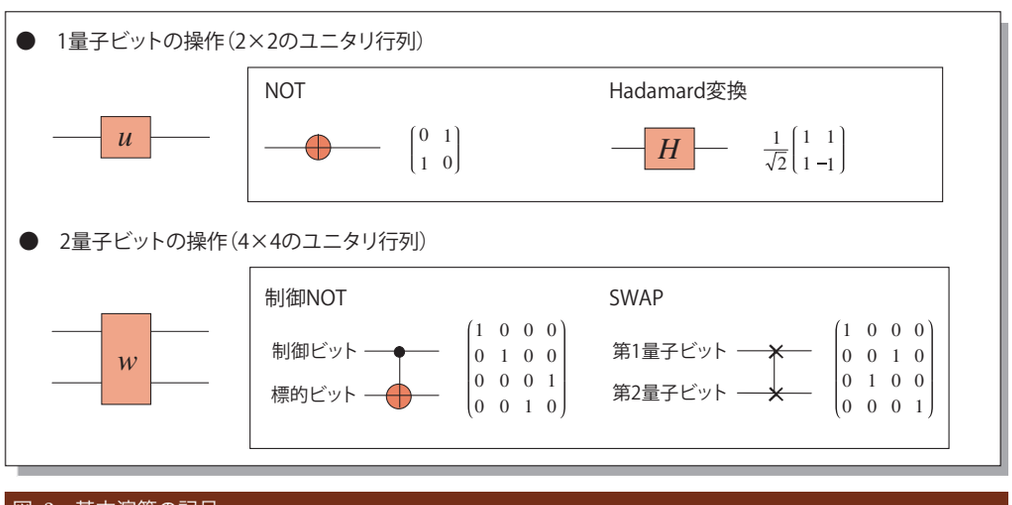

図-2　基本演算の記号

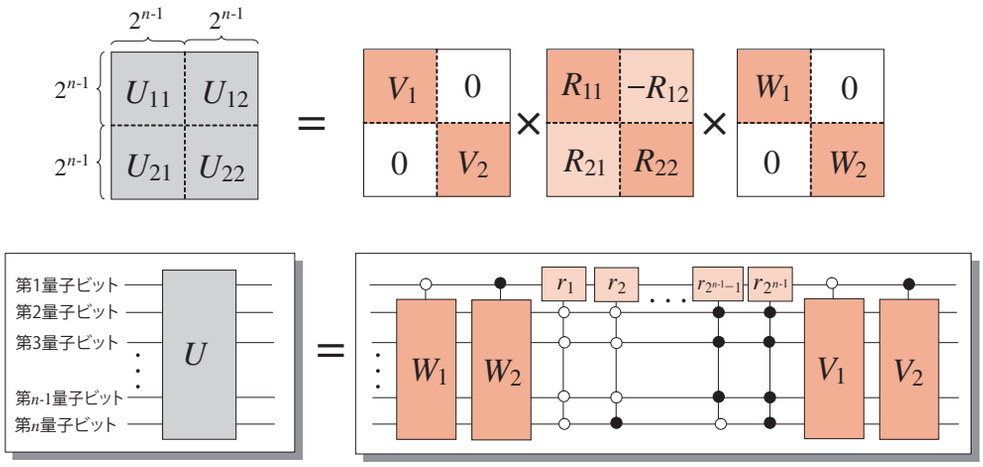

図 -3　Cosine-Sine 分解と対応する量子回路

クトルから別の 2^n 次元ベクトルへの写像を表す $2^n \times 2^n$ のユニタリ行列で書ける．また，入力状態を変化させない演算は，単位行列に対応する．

すると，図 -1 の量子回路は，以下の行列の積で書ける．

$$(u_3 \otimes u_4 \otimes I)\,(I \otimes C)\,(I \otimes u_1 \otimes u_2)(C \otimes I). \qquad (2)$$

ここで，I は 2×2 の単位行列，u_1, u_2, u_3, u_4 は 2×2 のユニタリ行列，C は制御 NOT で，図 -2 に示す 4×4 行列である．行列演算の性質から，量子回路における基本操作の順序（左から右）とは逆順に行列がかかっていることに注意する [6]．

このような行列の記述を利用すると，$2^n \times 2^n$ のユニタリ行列を，（2）のような基本演算に対応する行列の積へ変換することで，量子回路の設計ができることが分かる．次章では，行列の特異値分解を利用して，これを実現する手法を紹介する．

行列分解を用いた量子回路の自動設計

行列分解を用いた量子回路設計では，特異値分解や固有値分解が応用されている．ここでは，Cosine-Sine 分解という一般化特異値分解を求める手法を応用した量子回路の設計手法について述べる [2), 4)]．Cosine-Sine 分解は，行列を**図 -3** のような特徴を持った 3 つの行列の積に分解する手法で，入力の行列 U を 4 分割した各々の部分 U_{jk} について，$U_{jk} = V_j R_{jk} W_k$ $(j, k = 1, 2)$ が，それぞれ特異値分解となるような分解手法である．ここで，$R_{jk}\,(j, k = 1, 2)$ は，以下のような $2^{n-1} \times 2^{n-1}$ の対角行列である．

$$R_{11} = R_{22} = diag\,(\cos\theta_1, \cdots, \cos\theta_{2^{n-1}}),$$
$$R_{21} = R_{12} = diag\,(\sin\theta_1, \cdots, \sin\theta_{2^{n-1}}), \qquad (3)$$

Cosine-Sine 分解の 3 つの行列の積は，図 -3 量子回路に対応している．ここで，白丸（◦）の制御ビットは，制御ビットの状態が 0 のときに標的ビットにユニタリ行列を適用する制御演算で，黒丸（•）は，制御ビットの状態が 1 のときに標的ビットにユニタリ行列を適用する制御演算である．また，回路の中央部分に並んでいる制御ビットが $n-1$ 個の演算の列は，Cosine-Sine 分解における中央の行列に対応するもので，

$$r_k = \begin{pmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{pmatrix},\ k = 1,\ 2,\ \cdots, 2^{n-1}.$$

ここで，θ_k は (3) と対応している．

Cosine-Sine 分解の 3 つの行列の積のうち，両端の行列は制御ビットを無視すると，$n-1$ 量子ビットの演算 (V_1, V_2, W_1, W_2) であるため，再帰的に Cosine-Sine 分解を行うことができる．すると，**図 -4** のように，最終的に $n-1$ 個の制御ビットを持つ 1 量子ビットの演算の列にまで分解することができる．この演算は，図 -4 の (c) のような基本演算の列に分解できるので，最終的に入力のユニタリ行列に対応する基本演算の列（量子回路）を設計することができる．図 -4 の (b) から (c) への変換を求める場合には，$u_j\,(j = 1, \cdots, 2^{n-1})$ と $a_k\,(k = 1, \cdots, 2^{n-1} + 1)$ の行列間の対応関係を利用して，方程式を解くような形で基本演算の系列を求めることができる [2)]．なお，図 -4 では，半月型の制御ビットの記号を用いているが，これは (b) のような制御ビットが白と黒の場合で異なる演算を適用する演算を省略して記述したものである．

Cosine-Sine 分解が量子回路設計に適している理由は，図 -3 に示すように，分解後の行列が制御演算の積で表され，標的ビットの演算だけを見ると n 量子ビットよりも量子ビット数の少ない演算で書けることにある．このことにより，再帰的に Cosine-Sine 分解を適用することで，標的ビットの演算を 1 量子ビットの演算に近づけることができる．

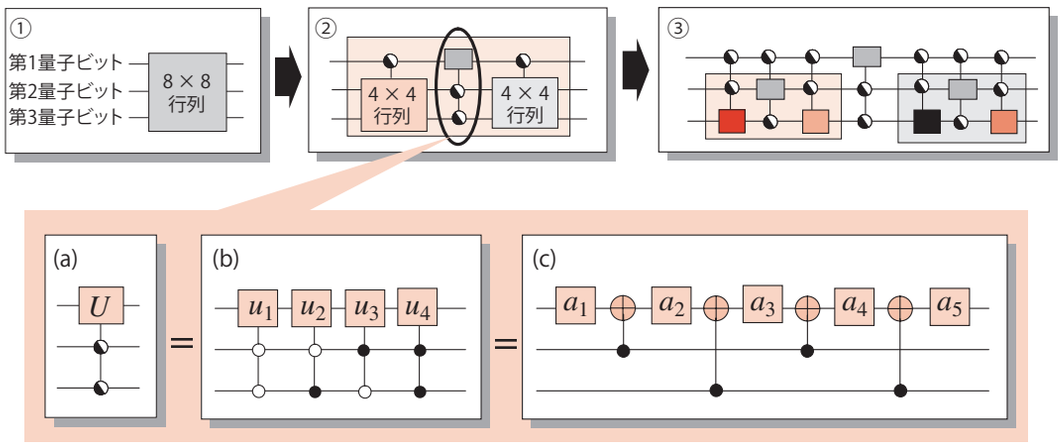

図 -4　Cosine-Sine 分解を再帰的に利用した量子回路設計

もっと一般的には，量子回路の設計に適した行列分解は，Lie 群論における KAK 分解という枠組みでとらえることができる．KAK 分解とは，ある $N \times N$ の行列からなる群 G に対して，$g \in G$ が次式のような 3 つの行列の積に分解できることを意味する．

$$g = k_2 a k_1,\ k_1, k_2 \in K, a \in A.$$

ここで，K, A は G の部分群である．Cosine-Sine 分解は KAK 分解の一例で，この場合，K は図 -3 の 3 つの行列の積の両端の行列で用いられているブロック対角行列からなる部分群，A は (3) を満たす行列からなる部分群である．Cosine-Sine 分解以外にも，量子回路設計のために考案された KAK 分解 (Khaneja-Glaser 分解) がある．Khaneja-Glaser 分解は，行列の対角化を応用して求めることができる [3]．

例：量子 Fourier 変換の量子回路の設計

❒ 量子 Fourier 変換とは

量子 Fourier 変換とは，出力の量子状態が，入力の量子状態の離散 Fourier 変換になるようなユニタリ変換である．n 量子ビットの入力と出力の量子状態をそれぞれ，

$$x = (x_0, x_1, \cdots, x_{2^n-1})^t,$$
$$y = (y_0, y_1, \cdots, y_{2^n-1})^t,$$

とおく．ここで，出力 y の各成分 $y_0, \cdots, y_{2^n-1}$ は，入力 x の離散 Fourier 変換

$$y_k = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \omega^{jk} x_j,\quad \omega = \exp\left(\frac{2\pi i}{2^n}\right),$$

となる $(k = 0, 1, \cdots, 2^n-1)$．このようなユニタリ変換 F_n は，その (k, j) 成分 $F_n[k, j]$ が

$$F_n[k, j] = \frac{1}{\sqrt{2^n}} \omega^{jk}, \qquad (4)$$

なる $2^n \times 2^n$ のユニタリ行列で書ける．

❒ 量子回路の設計

$\tilde{F}_n$ を以下の式で定義する．

$$\tilde{F}_n = \begin{pmatrix} F_{n-1} & DF_{n-1} \\ F_{n-1} & -DF_{n-1} \end{pmatrix}.$$

ここで，D は以下の対角行列である．

$$D = diag\left(\omega^0, \omega^1, \cdots, \omega^{2^{n-1}-1}\right).$$

$\tilde{F}_n$ は，量子 Fourier 変換 F_n の列を [奇数列，偶数列] の順に入れ替えることで得られる．たとえば，8×8 の量子 Fourier 変換の場合だと，列の順序を

$$[1,2,3,4,5,6,7,8] \rightarrow [1,3,5,7,2,4,6,8]$$

となるように入れ替えを行う．この列の入れ替えは，量子計算では量子ビットの入れ替え (SWAP) で実現できる．列の入れ替えを Q_n とおき，$F_n = \tilde{F}_n Q_n$ とする．$S_{(j,k)}$ を第 j 量子ビットと第 k 量子ビットの SWAP とすると，

$$Q_n = S_{(1,n)} S_{(2,n)} \cdots S_{(n-1,n)},$$

と書ける．

$\tilde{F}_n$ の Cosine-Sine 分解を考えると，**図 -5** のようになる．ここで，I_{n-1} は $2^{n-1} \times 2^{n-1}$ の単位行列を表す．F_{n-1} は $n-1$ 量子ビットの量子 Fourier 変換なので，再帰的にこの分解を繰り返す．残された基本演算でない要素は，第 1 量子ビットを制御ビットとして，$n-1$ 量子ビットに D を施す演算である．制御ビットを無視して，$n-1$ 量子ビットの演算 D について考えると，**図 -6** のように分解できる．ここで，d_k は，D の $(0, 0)$ 成分から $(2^{n-k}-1, 2^{n-k}-1)$ 成分までの要素からなる対角行列

$$d_k = diag\left(\omega^0, \omega^1, \cdots, \omega^{2^{n-k}-1}\right),$$

である．d_k は，図 -6 に示すように再帰的に分解していくと，最終的に $n-1$ 個の 1 量子ビットの基本演算列が求まる．ここで，

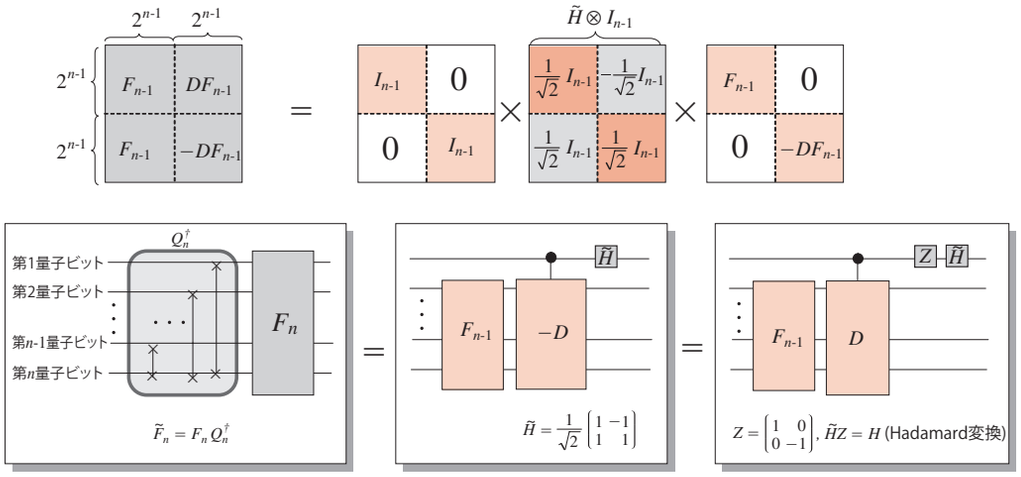

図 -5　量子 Fourier 変換 F_n の Cosine-Sine 分解と対応する量子回路

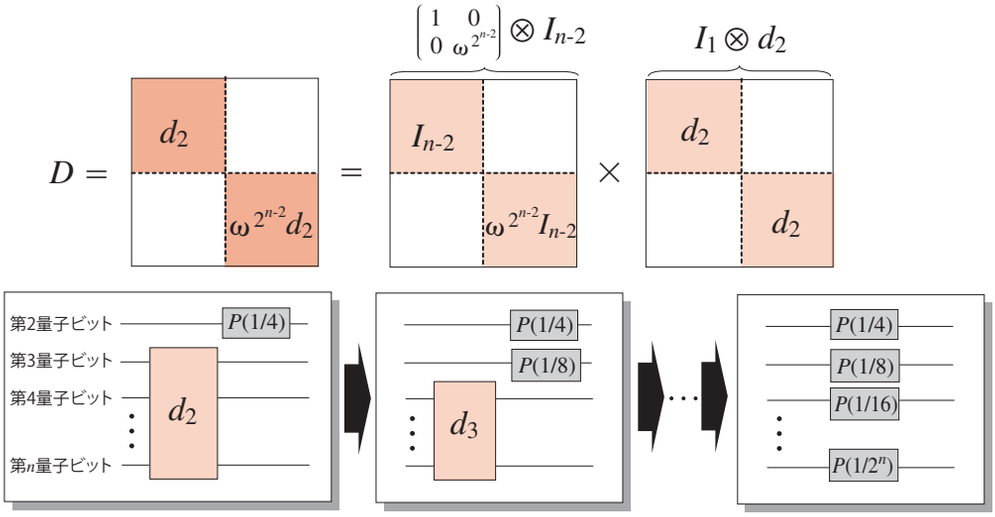

図 -6　対角行列 D の分解と対応する量子回路

$$P(\ell) = \begin{pmatrix}1 & 0\\ 0 & \exp(2\pi i \ell)\end{pmatrix}.$$

この 1 量子ビットの演算の列を図 -5 の D の部分に挿入すると，第 1 量子ビットを制御ビットとする演算の列が得られる．これらの結果を合わせると，4 量子ビットの場合は**図 -7** の回路が得られる．ここで，SWAP は列の交換に対応している．Q_n は $n-1$ 個の SWAP から構成されるため，4 量子ビットの場合だと，Q_2 Q_3 Q_4 の列に対応する 6 個の SWAP の列であるが，簡単化することで，2 個の SWAP で実現できる．一般に，F_n を分解すると，n 個の Hadamard 変換と $(n^2-n)/2$ 個の 2 量子ビットの演算（標的ビットのユニタリ行列が $P(\ell)$ である演算）と，$\lfloor n/2 \rfloor$ 個の SWAP の系列に分解できる．

ここでは簡単のため，量子 Fourier 変換に前処理として列の入れ替え $Q_n^\dagger$ を適用した行列 $\tilde{F}_n$ を用いて，効率的な回路を求めているが，Cosine-Sine 分解とは別の KAK 分解を利用すると，Q_n を事前に適用しなくても，効率的な量子回路を求めることができる [3]．

今後の課題

本稿では，特異値分解を利用した量子回路の設計手法について概説した．この手法は，n 量子ビットの量子計算の実現に必要な基本演算数や，制御 NOT の数の下限を求める研究に応用されている [2)〜4)]．

量子回路の設計手法は，n 量子ビットの量子計算を，量子コンピュータ上の実行命令の系列へ自動翻訳するコンパイラの要素技術としての応用も考えられるが，そのためには，計算コストや回路の最適化など多くの課題がある．

まず，ここで述べた量子回路の設計手法は特異値分解を利用するが，対象となる入力行列のサイズは，n 量子ビットに対して $2^n \times 2^n$ という大きな行列となるた

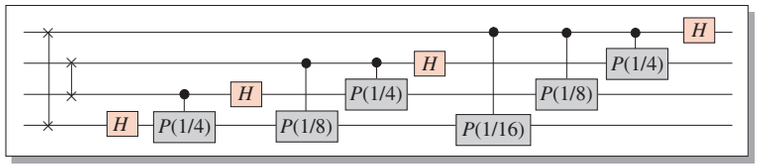

図 -7　4 量子ビットの量子 Fourier 変換の回路

め，計算コストが大きく，多量子ビットの演算にそのまま適用するのは難しい．また，量子 Fourier 変換は行列分解のみによって効率的な量子回路が自動生成できることを示したが，実際にはそれだけではなく，回路の最適化が必要な場合が多い．量子回路の最適化手法としては，これまでに Palindrome を用いた最適化手法 [5] や，制御 NOT のみで構成される量子回路を最適化する手法 [1] が提案されている．

基本演算として何を用いるか，という点も重要な問題である．本稿では，1 量子ビットの任意の演算と制御 NOT を基本演算としたが，1 量子ビットの演算は無限に存在する．したがって，基本演算としてこれらのすべてを実現するのは困難である．一方で，基本演算の集合として，あらかじめ有限個の基本演算を用意し，これらを組み合わせて任意のユニタリ演算を近似的に実現する方法が知られている（Solovay-Kitaev の定理）．近似を用いて効率的な量子回路を自動設計する手法を考案することも，今後の課題の 1 つである．

参考文献

1）Iwama, K., Kambayashi, Y. and Yamashita, S. : Transformation Rules for Designing CNOT-based Quantum Circuits, 39th Design Automation Conference, pp.419-424 (2002).
2）Möttönen, M., Vartiainen, J. J., Bergholm, V. and Salomaa, M. : Quantum Circuits for General Multiqubit Gates, Phys. Rev. Lett., Vol.93, No.13, p.130502 (2005).
3）Nakajima, Y., Kawano, Y. and Sekigawa, H. : An Algorithm for Producing Quantum Circuits Using KAK Decompositions, Quantum Inf. Comput., Vol.6, No.1, pp.67-80 (2006).
4）Shende, V. V., Markov, I. L. and Bullock, S. S. : Synthesis of Quantum Llogic Circuits, IEEE Trans. on Computer-Aided Design, Vol.25, No.6, pp.1000-1010 (2006).
5）Svore, K. M. and Aho, A. V. : The Design and Optimization of Quantum Circuits Using the Palindrome Transform, the ERATO Conference on Quantum Information Sciences (EQIS), Kyoto, Japan, pp.5-7 (2003). http://arxiv.org/abs/quant-ph/0311008
6）上坂吉則 : 量子コンピュータの基礎数理，コロナ社 (2000).

（平成 18 年 10 月 20 日受付）

中島　裕美（正会員）
nakajima@theory.brl.ntt.co.jp
2001 年愛媛大学工学部情報工学科卒業．2003 年同大学院博士前期課程修了．同年日本電信電話（株）入社．現在，同社 NTT コミュニケーション科学基礎研究所にて，量子情報処理，特に量子回路設計理論の研究に従事．

河野　泰人
kawano@theory.brl.ntt.co.jp
NTT コミュニケーション科学基礎研究所主任研究員．1989 年大阪大学理学部数学科卒業．1991 年名古屋大学大学院前期課程修了．同年日本電信電話（株）入社．博士（情報科学）．現在の主な研究テーマは量子情報理論．電子情報通信学会，日本ソフトウェア科学会各会員．

関川　　浩（正会員）
sekigawa@theory.brl.ntt.co.jp
1989 年東京大学大学院理学系研究科修士課程修了．同年日本電信電話（株）入社．2000 年から 2002 年まで西日本電信電話（株）勤務．ハードウェア設計支援技術の研究を経て，数式処理，数値–数式融合計算の研究に従事．博士（数理科学）．