Interpolation Methods for Binary and Multivalued Logical Quantum Gate Synthesis

Zeno Toffano, François Dubois

Interpolation Methods for Binary and Multivalued Logical Quantum Gate Synthesis

Zeno TOFFANO[a,b], François DUBOIS[c,d]

[a] Telecom Dept., CentraleSupélec, Gif-sur-Yvette, France
[b] Lab. des Signaux et Systèmes - L2S (UMR8506) - CNRS - U. Paris-Saclay, France
[c] LMSSC, CNAM, Paris, France
[d] Department of Mathematics, University Paris-Sud, Orsay, France
zeno.toffano@centralesupelec.fr ; francois.dubois@math.u-psud.fr

Abstract

A method for synthesizing quantum gates is presented based on interpolation methods applied to operators in Hilbert space. Starting from the diagonal forms of specific generating seed operators with non-degenerate eigenvalue spectrum one obtains for arity-one a family of logical operators corresponding to the one-argument logical connectives. Scaling-up to n-arity gates is obtained by using the Kronecker product and unitary transformations. The quantum version of the Fourier transform of Boolean function is presented and a method for Reed-Muller decomposition is derived. The common control gates can be easily obtained by considering the logical correspondence between the control logic operator and the binary logic operator. A new polynomial and exponential formulation of the Toffoli gate is presented. The method has parallels to quantum gate-T optimization methods using powers of multilinear operator polynomials. The method is then applied naturally to alphabets greater than two for multi-valued logical gates used for quantum Fourier transform, min-max decision circuits and multivalued adders.

[paper accepted at TQC2017, Paris, France, June 14-16, 2017].

Keywords: *quantum gates, linear algebra, interpolation, Boolean functions, quantum angular momentum, multivalued logic*

1 Introduction

Quantum calculation methods are becoming a strategic issue for emerging technologies such as quantum computing and quantum simulation. Because of the limited (at this date) quantum computational resources available it is very important to reduce the resources required to implement a given quantum circuit. The problem of optimal quantum circuit synthesis is an important topic much of the effort focuses on decomposition methods of logical gates. Quantum reversible gates have been extensively studied using principally Boolean functions implemented on Clifford, controlled-not and Toffoli gates and also non-Clifford such as the T gate. The advantages and drawbacks for privileging certain families of gates has been thoroughly investigated. Basically all methods rely on the control-logic paradigm as was first proposed by David Deutsch in [1]. The 2-qubit controlled-not gate being its simplest element resulting in the operation $|x, y > \rightarrow |x, x \oplus y >$ where the exclusive disjunction (XOR, $\oplus$) acts as logical negation on qubit y when the control qubit x is at one and leaves it unchanged otherwise. The universality of this logic is assured by the double-controlled-not or Toffoli gate with the operation on three qubits $|x, y, z > \rightarrow |x, y, xy \oplus z >$. This gate operates on the third qubit, z, when the conjunction is met on the control bits x and y (both must be 1) being thus equivalent to a negated binary conjunction $NAND$ gate, which is known to be universal. These logical controlled-gates transform the qubits as reversible permutation operators *i.e.* they are not diagonal in the computational basis.

In recent years many synthesizing methods for quantum circuits actually use operators that are diagonal in the computational basis because of the simpler resulting mathematical operations. Popular are the controlled-Z and the double-controlled-Z gates from which are easily built controlled-not and Toffoli gates, solutions using diagonal T and S gates, as will be shown hereafter, are also often proposed. Also the stabilizer formalism and the surface codes use families of Clifford diagonal gates. So in some way the "back to diagonal" trend seems to have some advantages and this also when considering traditional problems in quantum physics where most efforts are made to define Hamiltonians and find the corresponding energy spectrum (eigenvalues) and stationary states (eigenstates) which can be implemented in quantum simulation models.

A question arises: can logical calculations be formalized someway directly in the qubit eigenspace? The answer seems to be affirmative and in a rather simple way too. A proposal has been given by making parallels between propositional logic and operator linear algebra in the framework of "Eigenlogic" [2, 3], the idea is that Boolean algebra operations can be represented by operations using projection operators (but not only as will be shown hereafter) in Hilbert space.

A theoretical justification could be inspired by Pierre Cartier in [4], relating the link between the algebra of logical propositions and the set of all valuations on it: "...in the *theory of models* in logic a model of a set of propositions has the effect of validating certain propositions. With each logical proposition one can associate by duality the set of all its true valuations represented by the number 1. This correspondence makes it possible to interpret the algebra of propositions as a class of subsets, conjunction and disjunction becoming respectively the intersection and union of sets. This corresponds to the *Stone duality* proved by the Stone representation theorem and is one of the spectacular successes of twentieth century mathematics....The development of quantum theory led to the concept of a quantum state, which can be understood as a new embodiment of the concept of a valuation".

The idea is not new, and stems from John Von Neumann's proposal of "projections as propositions" in [5] which was subsequently formalized in quantum logic with Garret Birkhoff in [6].

But because quantum logic is mostly interested in aspects concerning non commuting operations going beyond and also in contrast with the principles of classical logic it is still not considered as an operational tool for quantum computing even though many bridges have been made [7, 8]. The aim of Eigenlogic is on the other side to fully exploit the logical structure offered by the operational system in the eigenspace with of course the possibility to look outside at other basis representations this has led for example to *fuzzy logic* applications in [3].

While most of the research is currently devoted to quantum circuit and algorithm developments based on the use of quantum operators working in binary (Boolean) systems through the manipulation of qubits, there are many possibilities for the exploitation of observables with more than two non-degenerate eigenvalues and the theory of multi-valued-logic is very naturally applicable to the design and analysis of these systems.

2 The interpolation method for quantum operators

Generalizing the method used in [2, 3] we look for n-arity logical operators supporting a finite set of m distinct logical values.

2.1 The Seed Operator and the associated projection operators

The method presented here is inspired from classical *Lagrange interpolation* where the "variable" is represented by a *seed operator* acting in Hilbert space, possessing m distinct eigenvalues, *i.e.* it is non-degenerate. The values are not fixed meaning that one can work with different alphabets, in the binary case one uses classically the Booleans $\{0, 1\}$, but also $\{+1, -1\}$ are often considered. What this method will show is that for whatever finite system of values, unique logical operators can be defined. In the multivalued case, popular choices are, for example, the system $\{0, 1, ..., m-1\}$ formalized in Post logic [9] and rational fractional values in the unit interval $[0, 1]$ giving the system$\{0, \frac{1}{m}, \frac{2}{m}..., \frac{m-1}{m}, 1\}$ as used in Łukasiewicz logic. Other convenient numerical choices are the balanced ternary values $\{+1, 0, -1\}$, these can characterize qutrits associated to quantum orbital angular momentum observables (see [3]).

When the eigenvalues are real the corresponding logical operators are *observables*, *e.g.* Hermitian operators. Many logical gates are of this type, the most popular ones are the Pauli-$\boldsymbol{Z}$, Pauli-$\boldsymbol{X}$ gates, the

Hadamard gate and the control gates: controlled-not and Toffoli. These observable gates are also named classical gates.

The possibility of using complex values can also be considered and this has applications for specific problems such as in T-gate synthesis and quantum Fourier transform in this last case one considers the roots of unity for m values giving the spectrum $\left\{e^{i2\pi\cdot 0}, e^{i2\pi\cdot\frac{1}{m}}, e^{i2\pi\cdot\frac{2}{m}}, ..., e^{i2\pi\cdot\frac{m-1}{m}}\right\}$.

In each situation one starts by defining the *seed operator* $\mathbf{\Lambda}$ with m non-degenerate eigenvalues λ_i. The density matrix of eigenstate $|\lambda_i>$ is $\mathbf{\Pi}_{\lambda_i} = |\lambda_i><\lambda_i|$, and corresponds to the rank-1 projection operator, *a ray*, for this state. It is obtained by the following expression:

$$\mathbf{\Pi}_{\lambda_i}(\mathbf{\Lambda}) = \prod_{j=1\,(j\neq i)}^{m} \frac{(\mathbf{\Lambda} - \lambda_i \mathbf{I}_m)}{(\lambda_i - \lambda_j)} \tag{1}$$

this expression corresponds to the *Duncan-Collar formula.* The development is unique according to the *Cayley-Hamilton theorem.* Expression (1) is polynomial consisting in a linear combination of powers up to $m-1$ of the seed operator $\mathbf{\Lambda}$, it is a rank-1 projection operator over the m dimensional Hilbert space and can be expressed as a $m \times m$ square matrix.

The number m of rays is the dimension of the vector space. All the rank-1 projection operators obtained in (1) for the eigensystem commute and span the entire space by the closure relation:

$$\sum_{i=1}^{m} \mathbf{\Pi}_{\lambda_i} = \sum_{i=1}^{m} |\lambda_i><\lambda_i| = \mathbf{I}_m \tag{2}$$

2.2 Logical operators for arity-one

As outlined in [2, 3] in Eigenlogic eigenvalues correspond to logical truth values (values $\{0, 1\}$ for a Boolean system) and one can make the correspondence between propositional logical connectives and operators in Hilbert space. The truth table of a logical connective corresponds to different semantic *interpretation*s, each interpretation is a fixed attribution of truth values to the elementary propositions (the "inputs") composing the connective. In Eigenlogic each interpretation corresponds to one of the eigenvectors and the associated eigenvalue to the corresponding truth-value for the considered logical connective.

It is well known that logical connectives can be expressed through arithmetic expressions, these are closely related to polynomial expressions over rings, but with expressions using ordinary arithmetic addition and subtraction instead of their modular counterpart as used in Boolean algebra. These topics were thoroughly discussed in [2] for the Boolean values $\{0, 1\}$. Arithmetic developments of logical connectives are often used, for example, in the description of switching functions for decision logic design. A good review was given by Svetlana Yanushkevich in [10].

In logic, the functions and their arguments take the same values and these are the only possible logical values. For Boolean functions these unique possible values are the two numbers 0 and 1 corresponding respectively to the *False* and *True* character of a logical proposition. So considering an arithmetic expression for an arity-one logical connective $\ell(p)$ and choosing the m distinct logical values a_i, the value taken by the logical function at one of these points a_p is $\ell(a_p) \in \{a_1, a_2, ..., a_m\}$, also one of these values. The corresponding unique logical operator is given by the interpolation development:

$$\boldsymbol{F}_\ell = \sum_{i=1}^{m} \ell(a_i)\, \mathbf{\Pi}_{a_i} \tag{3}$$

this matrix decomposition formula is proved by *Sylvester's theorem* and represents the *spectral decomposition* of the operator. The projection operators are obtained by equation (1) with $\lambda_i = a_i$. It has to be outlined that the method used for obtaining expression (3) is the counterpart for operators of the classical Lagrange interpolation method over m points [11].

One must then consider the operators corresponding to elementary propositions of the logical connectives. This is a straightforward procedure in Eigenlogic. For arity-one logical connectives, these are function of one single elementary proposition p which corresponds in the method proposed here to the seed operator $\mathbf{\Lambda}$. These concepts will become clearer with some examples presented hereafter.

2.3 Scaling up to higher arity

The scaling is obtained by following the same procedure as for classical interpolation methods for multivariate systems using tensor products (see *e.g.* [11]). Here the chosen convention for the indexes is the one given by David Mermin in [12] where index 0 indicates the lowest digit, index 1 the next and so on... For arity-2 one considers two operators $\boldsymbol{P}_1$ and $\boldsymbol{Q}_0$ corresponding to the propositional variables p and q. One can write by the means of the Kronecker product $\otimes$:

$$\boldsymbol{P}_1 = \boldsymbol{\Lambda} \otimes \mathbf{I} \qquad , \qquad \boldsymbol{Q}_0 = \mathbf{I} \otimes \boldsymbol{\Lambda} \tag{4}$$

for arity-3 using three operators one has:

$$\boldsymbol{P}_2 = \boldsymbol{\Lambda} \otimes \mathbf{I} \otimes \mathbf{I}\,, \qquad \boldsymbol{Q}_1 = \mathbf{I} \otimes \boldsymbol{\Lambda} \otimes \mathbf{I}\,, \qquad \boldsymbol{R}_0 = \mathbf{I} \otimes \mathbf{I} \otimes \boldsymbol{\Lambda} \tag{5}$$

for higher arity-n the procedure can be automatically iterated.

Logical operators can then be obtained by (3), and are multi-linear combinations of the elementary operators $\boldsymbol{P}_i$, $\boldsymbol{Q}_j$...., for example the simplest operators are products of these. Due to logical completeness (demonstrated by Emil Post in [9]) there will be for an m-valued n-arity system exactly m^{m^n} logical operators forming a complete family of commuting logical operators. Logical completeness has also another important consequence: there are always universal connectives from which all the others can be derived, it has been shown, also in [9], that for an m-valued arity-2 system one needs at least two universal connectives, in the case of Post logic these turn out to be the *general negation* (cyclic permutation of all values $a_i \to a_{i+1}$) and the Max connective (takes the highest of the two input values) these reduce to Boolean connectives for binary values where the Max connective becomes the disjunction (OR, $\vee$). The logical operators obtained here for an m-valued n-arity system are represented by $m^n \times m^n$ square matrices.

3 Interpolation synthesis of binary quantum gates

In quantum computing operations on qubits are done using reversible gates without reducing the dimension of the vector space, except at the end of the calculation where storage of some values is done by projection operators transforming the qubits into classical bits.

3.1 Fourier transform of Boolean functions and Reed-Muller decomposition for logical operators

The following methods are based on some basic mathematical properties of logical functions, that have their origin in George Boole's elective development theorem for logical functions [13, 2], in the Fourier analysis of Boolean functions based on the Walsh transform [14, 15] and on the Reed-Muller developments of Boolean functions [16, 17].

With the notations used for the *Fourier analysis of Boolean functions* method, a self-inverse logical operator ($\boldsymbol{G}^2 = \mathbf{I}$) of arity-$n$ can be expressed as a unique development:

$$\boldsymbol{G}_\ell^{[n]} = \sum_{S \subseteq [n]} \frac{1}{2^n} \left[\left(\sum_{p=0}^{2^n-1} (-1)^{\boldsymbol{p}\cdot\boldsymbol{s}} g_\ell^{[n]}(\boldsymbol{p}) \right) \boldsymbol{\chi}_S^{[n]} \right] = \sum_{p=0}^{2^n-1} g_\ell^{[n]}(\boldsymbol{p})\, \boldsymbol{\Pi}_{\boldsymbol{p}} \qquad , \quad g_\ell^{[n]} \in \{+1, -1\} \tag{6}$$

where $[n]$ is the *powerset* of the logical system and S a subset of the powerset, the sum is taken over all possible subsets, the n-bit string vector $\boldsymbol{s}$ characterizes the subset S. The function $g_\ell^{[n]} : \{+1, -1\}^n \to \{+1, -1\}$ represents the scalar logical function. The truth values are here $\{+1, -1\}$, representing respectively *False* and *True* and correspond to the Booleans $\{0, 1\}$. In (6) is also shown the interpolation form using the rank-1 projection operators $\boldsymbol{\Pi}_{\boldsymbol{p}}$ at the interpolation "point" represented by the n-bit string vector $\boldsymbol{p}$. The bitwise scalar product of the two strings, $\boldsymbol{p} \cdot \boldsymbol{s}$, determines the sign of each term of the development. The operators $\boldsymbol{\chi}_S^{[n]}$ are also self-inverse and local, they are function of the seed operator $\boldsymbol{Z} = diag_z(+1, -1)$:

$$\boldsymbol{\chi}_{S \neq [\emptyset]}^{[n]} = \prod_{k \in S \neq \emptyset} \boldsymbol{U}_k^{[n]}\,, \qquad \boldsymbol{U}_k^{[n]} = \mathbf{I}_2^{\otimes(n-k-1)} \otimes \boldsymbol{Z} \otimes \mathbf{I}_2^{\otimes k} \qquad , \boldsymbol{\chi}_{[\emptyset]}^{[n]} = \mathbf{I}_{2^n} \tag{7}$$

where $\boldsymbol{U}_k^{[n]}$ are named the logical *dictators* [14, 15]. There exists a linear bijection between the self-inverse operators $\boldsymbol{G}$ and the respective idempotent projection operator $\boldsymbol{\Pi}_{\boldsymbol{G}}$ given by the *Householder transform*:

$$\boldsymbol{G} = \mathbf{I} - 2\boldsymbol{\Pi}_{\boldsymbol{G}} = (-1)^{\boldsymbol{\Pi}_{\boldsymbol{G}}} = e^{i\pi\boldsymbol{\Pi}_{\boldsymbol{G}}} \qquad , \quad \boldsymbol{\Pi}_{\boldsymbol{G}} = \frac{\mathbf{I} - \boldsymbol{G}}{2} \qquad , \quad \boldsymbol{G} = e^{i\frac{\pi}{2}} e^{-i\frac{\pi}{2}\boldsymbol{G}} \tag{8}$$

A consequence of equations, (6), (7) and (8) is that all the operators of a given family $\boldsymbol{G}^{[n]}$ commute and every $\boldsymbol{G}^{[n]}$ commutes with every $\boldsymbol{\Pi}^{[n]}$. This means technically that all operations on exponential of matrices, as given by equation (8), are simple arithmetic operations. For example the product of exponentials of operators will be the exponential of the sum of the operators. For the logical operators $\boldsymbol{G}^{[n]}$ the eigenvalues are $+1$ and -1, these numbers represent the truth values of the logical system. Expression (6) can also be derived directly by the interpolation method and expressed as a function of the corresponding interpolation projection operators $\boldsymbol{\Pi}_{\boldsymbol{p}}^{[n]}$. Both expansions in $\boldsymbol{\chi}_S^{[n]}$ or $\boldsymbol{\Pi}_{\boldsymbol{p}}^{[n]}$ can be used when transforming operator forms, some examples will be illustrated hereafter. In general for an n-arity system one has a family of 2^{2^n} different commuting operators $\boldsymbol{G}_\ell^{[n]}$. Also from the last equation in (8) one has the simple but important result that the operators $\boldsymbol{G}$ defined as sums in (6) can always be tranformed in products of quantum gates, some examples will be given hereafter.

The Eigenlogic operators [3] for exclusive disjunction (XOR, $\oplus$) for an arity-2 and arity-n system are defined by:

$$\boldsymbol{G}_{XOR}^{[2]} = diag_z(+1, -1, -1, +1) = \boldsymbol{U}_1^{[2]} \cdot \boldsymbol{U}_0^{[2]} \,, \qquad \boldsymbol{G}_{\oplus^n}^{[n]} = \prod_{j=0}^{n-1} \boldsymbol{U}_{j-1}^{[n]} \tag{9}$$

so the function XOR is represented by the matrix product of the single qubit dictators defined in (7) this operator is local because it is a Kronecker product of local operators.

The corresponding logical projection operator $\boldsymbol{\Pi}_{XOR}$ is given by the polynomial expression (see [2, 10]) using $\boldsymbol{\Pi}_0^{[2]} = \mathbf{I}_2 \otimes \boldsymbol{\Pi}$ and $\boldsymbol{\Pi}_1^{[2]} = \boldsymbol{\Pi} \otimes \mathbf{I}_2$ one has :

$$\boldsymbol{\Pi}_{XOR}^{[2]} = \boldsymbol{\Pi}_1^{[2]} + \boldsymbol{\Pi}_0^{[2]} - 2\,\boldsymbol{\Pi}_1^{[2]} \cdot \boldsymbol{\Pi}_0^{[2]} = diag_z(0, 1, 1, 0) \tag{10}$$

According to equation (8) this linear combination becomes the argument of the exponent of (-1) for $\boldsymbol{G}_{XOR}^{[n]}$. One can thus change all the $-$'s into $+$'s, also all the terms multiplied by 2^n with $n > 0$ vanish and so:

$$\boldsymbol{G}_{XOR}^{[2]} = (-1)^{(\boldsymbol{\Pi}_1^{[2]} + \boldsymbol{\Pi}_0^{[2]} - 2\,\boldsymbol{\Pi}_1^{[2]} \cdot \boldsymbol{\Pi}_0^{[2]})} = (-1)^{(\boldsymbol{\Pi}_1^{[2]} + \boldsymbol{\Pi}_0^{[2]})} = (-1)^{\boldsymbol{\Pi}_1^{[2]}} \cdot (-1)^{\boldsymbol{\Pi}_0^{[2]}} = \boldsymbol{U}_1^{[2]} \cdot \boldsymbol{U}_0^{[2]} \tag{11}$$

giving again (9). Now one can consider explicitly the action on qubits, with $|x>$, $x \in \{0, 1\}$, representing a single qubit eigenstate in the computational basis. The state for arity-2, is $|xy>$ with $x, y \in \{0, 1\}$ and so on when scaling for higher arity. The application of the preceding operators for exclusive disjunction (XOR, $\oplus$) on the 2-qubit state $|xy>$ and the 3-qubit state $|xyz>$ gives then:

$$\boldsymbol{G}_{XOR}^{[2]}\,|xy> = \boldsymbol{G}_{x\oplus y}^{[2]}\,|xy> = (-1)^{x+y}|xy> \qquad , \qquad \boldsymbol{G}_{x\oplus y\oplus z}^{[3]}\,|xyz> = (-1)^{x+y+z}|xyz> \tag{12}$$

For conjunction (AND, $\wedge$) logical operators one has for a 3-qubit state $|xyz>$:

$$\boldsymbol{\Pi}_{x\wedge y\wedge z}^{[3]}|xyz> = (xyz)\,|xyz> \qquad , \qquad \boldsymbol{G}_{x\wedge y\wedge z}^{[3]}\,|xyz> = (-1)^{xyz}|xyz> \tag{13}$$

Knowing the polynomial arithmetic expression of the logical connective one can derive the Reed-Muller form [10, 16, 17]. An important example is the disjunction operator (OR, $\vee$). One starts from the arithmetic expansion for the disjunction connective, which is actually the *inclusion-exclusion* rule, for arity-3 this gives:

$$x \vee y \vee z = x + y + z - xy - xz - yz + xyz \tag{14}$$

giving

$$\boldsymbol{G}_{x\vee y\vee z}^{[3]}\,|xyz> = (-1)^x(-1)^y(-1)^z(-1)^{xy}(-1)^{xz}(-1)^{yz}(-1)^{xyz}|xyz> \tag{15}$$

and the operator form:

$$\boldsymbol{G}^{[3]}_{x\vee y\vee z} = \boldsymbol{U}^{[3]}_2 \cdot \boldsymbol{U}^{[3]}_1 \cdot \boldsymbol{U}^{[3]}_0 \cdot \boldsymbol{G}^{[3]}_{x\wedge y} \cdot \boldsymbol{G}^{[3]}_{x\wedge z} \cdot \boldsymbol{G}^{[3]}_{y\wedge z} \cdot \boldsymbol{G}^{[3]}_{x\wedge y\wedge z} \tag{16}$$

which corresponds to the Reed-Muller decomposition of the disjunction on Booleans:

$$x \vee y \vee z = x \oplus y \oplus z \oplus (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \oplus (x \wedge y \wedge z) \tag{17}$$

3.2 Permutation operators and Pauli gates

Permutation operators $\boldsymbol{P}$ are unitary operators. Many quantum gates are permutation operators, *e.g:* Pauli-$\boldsymbol{X}$, *controlled-not*, *swap* and *Toffoli*, which are also self-inverse, *e.g.* $\boldsymbol{P} = \boldsymbol{P}^{-1}$ and thus $\boldsymbol{P^2} = \mathbf{I}$, meaning that their eigenvalues are $+1$ and -1 and one can apply the method shown above to obtain their polynomial operator form. Other operators have the same eigenvalues, this is the case of the Pauli gate group generated by the three Pauli operators, $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ and the identity operator $\mathbf{I}$ and combined by the Kronecker product. In the computational basis the $\boldsymbol{Z}$ and $\boldsymbol{X}$ and Hadamard $\boldsymbol{H}$ gate's matrix forms are:

$$\boldsymbol{Z} = \begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix} \quad , \quad \boldsymbol{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad \boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & 1 \\ 1 & -1 \end{pmatrix} \tag{18}$$

with the transformation $\boldsymbol{X} = \boldsymbol{H} \cdot \boldsymbol{Z} \cdot \boldsymbol{H}$ and because $\boldsymbol{H}^2 = \mathbf{I}$ also $\boldsymbol{Z} = \boldsymbol{H} \cdot \boldsymbol{X} \cdot \boldsymbol{H}$:

The seed operator for the system $\{+1, -1\}$ in the computational basis is $\boldsymbol{\Lambda}_{\{+1,-1\}} = \boldsymbol{Z} = diag_z(+1, -1)$.

Using (1) one obtains the two projection operators on the qubit states $|0 >= \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1 >= \begin{pmatrix} 0 \\ 1 \end{pmatrix}$:

$$\boldsymbol{\Pi}_{+1}(\boldsymbol{Z}) = \frac{1}{2}(\mathbf{I} + \boldsymbol{Z}) = |0 >< 0| \qquad , \quad \boldsymbol{\Pi}_{-1}(\boldsymbol{Z}) = \frac{1}{2}(\mathbf{I} - \boldsymbol{Z}) = |1 >< 1| = \boldsymbol{\Pi} \tag{19}$$

The same method can be used when the operators are not diagonal in the computational basis, for example one could have chosen the $\boldsymbol{X}$ gate as the seed operator leading to similar expressions as in (19) by changing $\boldsymbol{Z}$ into $\boldsymbol{X}$.

The seed operator and the two projection operators permit to write the four arity-one logical operators: $\boldsymbol{Z}$ (*Dictator*), $-\boldsymbol{Z}$ (*Negation*), $+\mathbf{I}$ (*Contradiction*) and $-\mathbf{I}$ (*Tautology*) (see [2, 3] for a detailed discussion), the truth values $\{+1, -1\}$ corresponding respectively to *False* and *True*.

Before continuing for higher arity it is interesting to analyze the logical operator system corresponding to the Boolean values $\{0, 1\}$, it is straightforward to see that in this case the seed operator, $\boldsymbol{\Lambda}_{\{0,1\}}$, is the projection operator $\boldsymbol{\Pi}_1 = \boldsymbol{\Pi}$ given in (19), the other projection operator $\boldsymbol{\Pi}_0$ being the complement:

$$\boldsymbol{\Lambda}_{\{0,1\}} = diag_z(0, 1) = \boldsymbol{\Pi}_1 = \boldsymbol{\Pi} = |1 >< 1| \qquad , \qquad \boldsymbol{\Pi}_0 = \mathbf{I} - \boldsymbol{\Pi} = |0 >< 0| \tag{20}$$

As shown above in (8) there are interesting relations for the operators when going from the system $\{+1, -1\}$ to the system $\{0, 1\}$. Due to the properties of the operator $\boldsymbol{\Pi}$ (idempotent projection operator $\boldsymbol{\Pi}^2 = \boldsymbol{\Pi}$) and $\boldsymbol{Z}$ (self-inverse operator $\boldsymbol{Z}^2 = \mathbf{I}$) one can use the Householder transform:

$$\boldsymbol{Z} = \mathbf{I} - 2\boldsymbol{\Pi} = (-1)^{\boldsymbol{\Pi}} = e^{i\pi\boldsymbol{\Pi}} = e^{+i\frac{\pi}{2}} e^{-i\frac{\pi}{2}\boldsymbol{Z}} \quad , \qquad \boldsymbol{Z}\,|x >= (-1)^x|x > \tag{21}$$

As shown above in (8) this transform is also valid for composite expressions and permits to transform operators with eigenvalues $\{0, 1\}$ into operators with eigenvalues $\{+1, -1\}$ and will be used hereafter to build the controlled-not end Toffoli gates.

For arity-2, one considers the two elementary operators acting on qubit-0 and qubit-1:

$$\boldsymbol{Z}_0 = \mathbf{I} \otimes \boldsymbol{Z} \qquad , \quad \boldsymbol{Z}_1 = \boldsymbol{Z} \otimes \mathbf{I} \tag{22}$$

The controlled-z gate, named here $\boldsymbol{C}_Z$, is diagonal in the computational basis and corresponds to a two-argument conjunction (AND, $\wedge$) in Eigenlogic. The interpolation method described above and used in [3] gives directly the known [12] polynomial expression:

$$\boldsymbol{C}_Z = diag_z(+1, +1, +1, -1) = \frac{1}{2}(\mathbf{I} + \boldsymbol{Z}_1 + \boldsymbol{Z}_0 - \boldsymbol{Z}_1 \cdot \boldsymbol{Z}_0) \tag{23}$$

3.3 Building the controlled-not and Toffoli gates

One can consider some interesting interpretations using Eigenlogic [3]: the truth table, in the alphabet $\{+1, -1\}$, given by the structure of the eigenvalues of the $\boldsymbol{C}_Z$ operator corresponds to the logical connective conjunction (AND, $\wedge$). The logical operator for conjunction $\mathbf{\Pi}_{AND}$ in the Boolean alphabet $\{0, 1\}$, as demonstrated in [2], has a very simple form:

$$\mathbf{\Pi}_{AND} = \mathbf{\Pi} \otimes \mathbf{\Pi} = diag_z(0,0,0,1) \quad , \qquad \mathbf{\Pi}_{AND}\,|xy> = xy\,|xy> \tag{24}$$

when applying this operator to the state $|xy>$ it gives the eigenvalue 1 only for the state $|11>$. One can transform this operator using the Householder transform in:

$$\boldsymbol{Z}_{AND} = \mathbf{I} - 2\mathbf{\Pi}_{AND} = (-1)^{\mathbf{\Pi}\otimes\mathbf{\Pi}} = diag_z(1,1,1,-1) = \boldsymbol{C}_Z \tag{25}$$

$$\boldsymbol{Z}_{AND}\,|xy> = \boldsymbol{C}_Z\,|xy> = (-1)^{xy}\,|xy> \tag{26}$$

justifying the interpretation of the $\boldsymbol{C}_Z$ gate as a conjunction in Eigenlogic.

The controlled-not operator, named here $\boldsymbol{C}$, can be expressed straightforwardly, see *e.g.* [12], in its polynomial form on $\boldsymbol{Z}_1$ and $\boldsymbol{X}_0$:

$$\boldsymbol{C} = \frac{1}{2}(\mathbf{I} + \boldsymbol{Z}_1 + \boldsymbol{X}_0 - \boldsymbol{Z}_1 \cdot \boldsymbol{X}_0) \tag{27}$$

this expression is derived from (23) using the transformation $\boldsymbol{X}_0 = \boldsymbol{H}_0 \cdot \boldsymbol{Z}_0 \cdot \boldsymbol{H}_0$, with $\boldsymbol{H}_0 = \mathbf{I} \otimes \boldsymbol{H}$ and where $\boldsymbol{H}$ is the Hadamard gate defined in (18).

In the preceding section it has been emphasized that one can consider different basis to define the system and the respective seed operator. Some important properties about projection operators (not necessarily commuting) have to outlined:

(i) The Kronecker product of two projection operators is also a projection operator.

(ii) If projection operators are rank-1 (a single eigenvalue is 1 all the others are 0) then their Kronecker product is also a rank-1 projection operator.

So for example one can use the computational basis ($\boldsymbol{Z}$ eigenbasis) for one qubit and the $\boldsymbol{X}$ eigenbasis for the other qubit. This is used for expressing the operator form of the controlled-not $\boldsymbol{C}$ gate as function of the projection operators. Taking the analogy with the $\boldsymbol{C}_Z$ gate one can define the projection operator associated to the $\boldsymbol{C}$ gate as:

$$\mathbf{\Pi}_C = \mathbf{\Pi} \otimes \mathbf{\Pi}_X = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \tag{28}$$

where the projection operator on the $\boldsymbol{X}$ eigenbasis is easily derived using $\mathbf{\Pi}_X = \boldsymbol{H} \cdot \mathbf{\Pi} \cdot \boldsymbol{H}$. Then one obtains straightforwardly:

$$\boldsymbol{C} = \mathbf{I} - 2\mathbf{\Pi}_C = (-1)^{\mathbf{\Pi}\otimes\mathbf{\Pi}_X} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{29}$$

It is also simple to derive expression (27) from (29).

The same method can be applied to build the doubly-controlled not-gate (Toffoli gate), named here $\boldsymbol{TO}$. One starts again with the analogy with conjunction and notices that the gate uses 3 qubits in a 8-dimensional space, the compound state in the computational basis is $|xyz>$. In the same way as before one can define here a doubly-controlled-Z gate $\boldsymbol{C}_{\boldsymbol{C}_Z}$:

$$\boldsymbol{C}_{\boldsymbol{C}_Z} = \mathbf{I} - 2(\mathbf{\Pi} \otimes \mathbf{\Pi} \otimes \mathbf{\Pi}) = (-1)^{\mathbf{\Pi}\otimes\mathbf{\Pi}\otimes\mathbf{\Pi}} = diag_z\,(1,1,1,1,1,1,1,-1)\ ,\ \boldsymbol{C}_{\boldsymbol{C}_Z}|xyz> = (-1)^{xyz}|xyz> \tag{30}$$

Then the $\boldsymbol{TO}$ gate can be found by the same method as for the $\boldsymbol{C}$ gate. The polynomial expression is easily calculated giving:

$$\begin{aligned} \boldsymbol{TO} &= \boldsymbol{H}_0 \cdot \boldsymbol{C}_{\boldsymbol{C}_Z} \cdot \boldsymbol{H}_0 = \mathbf{I} - 2(\boldsymbol{\Pi} \otimes \boldsymbol{\Pi} \otimes \boldsymbol{\Pi}_X) && (31) \\ &= \frac{1}{2} \cdot (\mathbf{I} + \boldsymbol{Z}_2 + \boldsymbol{C} - \boldsymbol{Z}_2 \cdot \boldsymbol{C}) && (32) \end{aligned}$$

The $\boldsymbol{C}$ gate can be expanded in equation (32) using equation (27) giving the alternative expression as function of single qubit gates:

$$\boldsymbol{TO} = \frac{1}{4}\left(3\mathbf{I} + \boldsymbol{Z}_2 + \boldsymbol{Z}_1 + \boldsymbol{X}_0 - \boldsymbol{Z}_2 \cdot \boldsymbol{Z}_1 - \boldsymbol{Z}_2 \cdot \boldsymbol{X}_0 - \boldsymbol{Z}_1 \cdot \boldsymbol{X}_0 + \boldsymbol{Z}_2 \cdot \boldsymbol{Z}_1 \cdot \boldsymbol{X}_0\right) \tag{33}$$

In quantum circuits it is practically difficult to realize the sum of operators and one prefers, if it is possible, to use a product form representing the same operator. Using the self-inverse symmetry of the above operators it is a standard procedure to make this transformation [12] using (8), a simple example is given by the transform (21). The $\boldsymbol{C}_Z$ gate polynomial expression in equation (23) can be transformed using (8), (23) and (25) in the following way:

$$\begin{aligned} \boldsymbol{C}_Z &= (-1)^{\boldsymbol{\Pi}_{\boldsymbol{C}_Z}} = e^{i\pi\boldsymbol{\Pi}_{\boldsymbol{C}_Z}} = e^{i\frac{\pi}{2}} e^{-i\frac{\pi}{2}\left[\frac{1}{2}(\mathbf{I}+\boldsymbol{Z}_1+\boldsymbol{Z}_0-\boldsymbol{Z}_1\cdot\boldsymbol{Z}_0)\right]} \\ &= e^{+i\frac{\pi}{4}} \left(e^{-i\frac{\pi}{4}\boldsymbol{Z}_1}\right) \cdot \left(e^{-i\frac{\pi}{4}\boldsymbol{Z}_0}\right) \cdot \left(e^{+i\frac{\pi}{4}\boldsymbol{Z}_1\cdot\boldsymbol{Z}_0}\right) && (34) \end{aligned}$$

The factorization of the exponential operators is allowed because all the argument operators in the exponential commute, the order of the multiplication can thus be interchanged.

The same method can be used to obtain an expression for the controlled-not $\boldsymbol{C}$ gate, one just replaces $\boldsymbol{Z}_0$ by $\boldsymbol{X}_0$.

In the same way this leads to a new factorized expression of the Toffoli gate $\boldsymbol{TO}$:

$$\begin{aligned} \boldsymbol{TO} &= e^{+i\frac{\pi}{8}} \cdot \left(e^{-i\frac{\pi}{8}\boldsymbol{Z}_2}\right) \cdot \left(e^{-i\frac{\pi}{8}\boldsymbol{Z}_1}\right) \cdot \left(e^{-i\frac{\pi}{8}\boldsymbol{X}_0}\right) \cdot \\ &\cdot \left(e^{+i\frac{\pi}{8}\boldsymbol{Z}_2\cdot\boldsymbol{X}_0}\right) \cdot \left(e^{+i\frac{\pi}{8}\boldsymbol{Z}_2\cdot\boldsymbol{Z}_1}\right) \cdot \left(e^{+i\frac{\pi}{8}\boldsymbol{Z}_1\cdot\boldsymbol{X}_0}\right) \cdot \left(e^{-i\frac{\pi}{8}\boldsymbol{Z}_2\cdot\boldsymbol{Z}_1\cdot\boldsymbol{X}_0}\right) && (35) \end{aligned}$$

this formulation shows also that it is easy to scale up the gates for example with a Toffoli-4 gate using three control bits on a 4 qubit state $|xyzw>$.

3.4 Correspondence with recent T-gate based methods

There has been much interest recently for developing general methods for synthesizing quantum gates based on polynomial methods [17, 18, 19]. The decomposition of arbitrary gates into Clifford and $\boldsymbol{T}$-set gates is an important problem. It is often desirable to find decompositions that are optimal with respect to a given cost function. The exact cost function used is application dependent; some possibilities are: the total number of gates; the total number of T -gates; the circuit depth and/or the number of ancillas used.

The single-qubit non-Clifford gate $\boldsymbol{T}$ and Clifford gate $\boldsymbol{S}$ are derived from the $\boldsymbol{Z}$ gate and are expressed in the computational basis in their matrix form:

$$\boldsymbol{T} = \boldsymbol{Z}^{\frac{1}{4}} = \begin{pmatrix} +1 & 0 \\ 0 & e^{+i\frac{\pi}{4}} \end{pmatrix} \quad , \quad \boldsymbol{S} = \boldsymbol{Z}^{\frac{1}{2}} = \begin{pmatrix} +1 & 0 \\ 0 & e^{+i\frac{\pi}{2}} \end{pmatrix} \tag{36}$$

as stated above these operators are diagonal in the computational basis.

Considering the preceding discussion the seed operator $\boldsymbol{\Lambda}$ for a $\boldsymbol{T}$-set is the gate itself with the eigenvalues $\{+1, \omega\}$, naming $\omega = e^{+i\frac{\pi}{4}}$.

The action of the $\boldsymbol{T}$-gate on a quibit in the computational basis is: $\boldsymbol{T}\,|x>= \omega^x\,|x>$. One can also define the conjugate transpose gate $\boldsymbol{T}^\dagger|x>= (\omega^\dagger)^x\,|x>= e^{-i\frac{\pi}{4}x}|x>= \omega^{-x}\,|x>$.

Using the two following arithmetic expressions of exclusive disjunction, (XOR, $\oplus$), for 2 and 3 Boolean arguments [2]:

$$x \oplus y \oplus z = x + y + z - 2xy - 2xz - 2yz + 4xyz \quad , \qquad x \oplus y = x + y - 2xy \tag{37}$$

where the second member is an inclusion-exclusion-like form. Combining the expressions gives:

$$4xyz = x + y + z - x \oplus y - x \oplus z - y \oplus z + x \oplus y \oplus z \tag{38}$$

this last expression gives a method for building more complex gates using only $\boldsymbol{T}$ and $\boldsymbol{T}^\dagger$ gates as shown by Peter Selinger in [18].

Starting again with the double controlled-Z gate $\boldsymbol{C_{C_Z}}$ one uses the fact, see (21), that $\boldsymbol{T}^4 = \boldsymbol{Z}$ and thus $\boldsymbol{T} = \boldsymbol{Z}^{\frac{1}{4}} = e^{i\frac{\pi}{8}} e^{-i\frac{\pi}{8}\boldsymbol{Z}}$. One defines the 3-qubit operators: $\boldsymbol{T}_0 = \mathbf{I} \otimes \mathbf{I} \otimes \boldsymbol{T}$, $\boldsymbol{T}_1 = \mathbf{I} \otimes \boldsymbol{T} \otimes \mathbf{I}$ and $\boldsymbol{T}_2 = \boldsymbol{T} \otimes \mathbf{I} \otimes \mathbf{I}$. Using (38) and the Reed-Muller methods discussed in section 3.1 one can express the $\boldsymbol{C_{C_Z}}$ operator using the action on the 3-qubit state $|xyz>$ as defined in (30) and the eigenvalue relation $(-1)^x = \omega^{4x}$:

$$\boldsymbol{C_{C_Z}} = \boldsymbol{T}_0 \cdot \boldsymbol{T}_1 \cdot \boldsymbol{T}_2 \cdot (\boldsymbol{T}^{[3]}_{x \oplus y})^\dagger \cdot (\boldsymbol{T}^{[3]}_{x \oplus z})^\dagger \cdot (\boldsymbol{T}^{[3]}_{y \oplus z})^\dagger \cdot (\boldsymbol{T}^{[3]}_{x \oplus y \oplus z}) \tag{39}$$

the operators corresponding to exclusive disjunction $\oplus$ in (39) can be easily obtained using the Eigenlogic interpretation: they are diagonal operators where the diagonal elements are the truth values of the corresponding logical connective using the $\boldsymbol{T}$ operator alphabet $\{+1, \omega\}$. For example explicitly:

$$\boldsymbol{T}^{[3]}_{x \oplus y} = \boldsymbol{T}^{[2]}_{x \oplus y} \otimes \mathbf{I} = diag_z(1, \omega, \omega, 1) \otimes \mathbf{I} = diag_z(1, 1, \omega, \omega, \omega, \omega, 1, 1) \tag{40}$$

It can be shown, again because $\boldsymbol{T} = \boldsymbol{Z}^{\frac{1}{4}} = e^{+i\frac{\pi}{8}} e^{-i\frac{\pi}{8}\boldsymbol{Z}}$, that the Toffoli gate $\boldsymbol{TO} = \boldsymbol{H}_0 \cdot \boldsymbol{C_{C_Z}} \cdot \boldsymbol{H}_0$ obtained using (39) is equivalent to expression (35). The operator given in (39) can be explicitly designed using the methods described in [18].

An alternative polynomial expression can be found directly by the interpolation method. The idea is that because $\boldsymbol{T}$ and $\boldsymbol{Z}$ commute and are not degenerate they have the same eigenvectors and thus one can use the same projection operators which are $\boldsymbol{\Pi}$ and its complement $\mathbf{I} - \boldsymbol{\Pi}$ (see equation (19)). The expression of the double controlled-Z gate $\boldsymbol{C_{C_Z}}$ as a function of $\boldsymbol{\Pi}$ was already calculated in (30), now one just has to express the projection operator $\boldsymbol{\Pi}$ as function of the operator $\boldsymbol{T}$ using (1), this gives:

$$\boldsymbol{\Pi}_\omega(\boldsymbol{T}) = \boldsymbol{\Pi} = (\omega - 1)^{-1}(\boldsymbol{T} - \mathbf{I}) \qquad , \qquad \boldsymbol{\Pi}_{+1}(\boldsymbol{T}) = \mathbf{I} - \boldsymbol{\Pi} = -(\omega - 1)^{-1}(\boldsymbol{T} - \omega\mathbf{I}) \tag{41}$$

so using directly (30) one has:

$$\boldsymbol{C_{C_Z}} = \mathbf{I} - 2(\omega - 1)^{-3} \left[(\boldsymbol{T} - \mathbf{I}) \otimes (\boldsymbol{T} - \mathbf{I}) \otimes (\boldsymbol{T} - \mathbf{I}) \right] \tag{42}$$

which can also be expressed as a function of $\boldsymbol{T}_0$, $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$:

$$\boldsymbol{C_{C_Z}} = \mathbf{I} + \frac{2}{(\omega - 1)^3} \left(\mathbf{I} - \boldsymbol{T}_2 - \boldsymbol{T}_1 - \boldsymbol{T}_0 + \boldsymbol{T}_2 \cdot \boldsymbol{T}_1 + \boldsymbol{T}_2 \cdot \boldsymbol{T}_0 + \boldsymbol{T}_1 \cdot \boldsymbol{T}_0 - \boldsymbol{T}_2 \cdot \boldsymbol{T}_1 \cdot \boldsymbol{T}_0 \right) \tag{43}$$

again using the transformation method it is easy to show that (43) is equivalent to (39). The Toffoli gate $\boldsymbol{TO}$ is also straightforwardly derived by replacing $\boldsymbol{T}_0$ by $(\boldsymbol{H}_0 \cdot \boldsymbol{T}_0 \cdot \boldsymbol{H}_0)$ in (43) leading again to (35).

The same method could be employed using $\boldsymbol{S}$ gates, for example simply by replacing the $\boldsymbol{T}$ operators by the respective $\boldsymbol{S}$ ones and $\omega = e^{+i\frac{\pi}{4}}$ by $\omega_S = e^{+i\frac{\pi}{2}}$ in (43). Also replacing ω by -1 and $\boldsymbol{T}$ by $\boldsymbol{Z}$ in (43) leads to an Eigenlogic operator expression for the three-input conjunction $\boldsymbol{C_{C_Z}} = \boldsymbol{G}^{[3]}_{x \wedge y \wedge z}$ as defined in (13).

4 Interpolation synthesis of multivalued quantum gates

Multi-valued logic requires a different algebraic structure than ordinary binary-valued one. Many properties of binary logic do not support set of values that do not have cardinality 2^n. Multi-valued logic is often used for the development of logical systems that are more expressive than Boolean systems for reasoning [19]. Particularly three and four valued systems, have been of interest with applications in digital circuits and computer science. Quantum physics and modern theory of many-valued logic were born nearly simultaneously [9] in the second and third decade of the twentieth century. The recently observed revival of interest in applying many-valued logic to the description of quantum phenomena is closely connected with fuzzy logic

[3]. Multivalued logic is of interest to engineers involved in various aspects of information technology. It has a long history of use in CAD with HDL (Hardware Description Languages) for simulation of digital circuits and their synthesis, vaqrious standards have been established. The total number of logical connectives for for an m-valued n-arity system is the combinatorial number m^{m^n} [9]. In particular for an arity-1 system with 3 values the number of connectives will be $3^{3^1} = 27$ and for an arity-2 system the number of connectives will be $3^{3^2} = 19683$. So it is clear that by increasing the values from 2 to m the possibilities of new connectives becomes intractable for a complete description of a logical system, but some special connectives play important roles and will be illustrated hereafter.

4.1 Multivalued operators for quantum Fourier transform

Following David Mermin in [12] one can introduce the following unitary operator $\mathfrak{S}$ in a 2^ndimension Hilbert space. Its action on the computational basis for a n-qubit state $|\boldsymbol{p}^{[n]}>$ is:

$$\mathfrak{S}|\boldsymbol{p}^{[n]}>= e^{-\frac{2i\pi}{2^n}d_p}|\boldsymbol{p}^{[n]}> \qquad , \quad (\mathfrak{S})^{2^n} = \mathbf{I}_{2^n} \tag{44}$$

where $|\boldsymbol{p}^{[n]}>=|p_{n-1}, ..., p_0>$ is the compound quantum state and $0 \le d_p \le 2^n - 1$ is the (decimal) number corresponding to the register of the state with digits $p_i \in \{0, 1\}$. The cyclic character of this operator is also showed.

The unitary operator $\boldsymbol{U}_{FT}$ corresponding to the quantum Fourier transform operation is then defined by:

$$\boldsymbol{U}_{FT}|\boldsymbol{q}^{[n]}>= \frac{1}{\sqrt{2^n}}\sum_{d_p=0}^{2^n-1} e^{-\frac{2i\pi}{2^n}d_q d_p}|\boldsymbol{p}^{[n]}>= (\mathfrak{S})^{d_q}\cdot \boldsymbol{H}^{\otimes n}|\mathbf{0}^{[n]}> \tag{45}$$

where $|\mathbf{0}^{[n]}>$ is the vector with all qubit digits at zero, $\boldsymbol{H}^{\otimes n}$ is the operator obtained by the Kronecker product of the n Hadamard operators (18).

The $\mathfrak{S}$ operator has a non-degenerate finite spectrum of eigenvalues, the roots of unity, and can thus be considered as a multivalued seed operator. An example can be illustrated where the seed projector can be directly written as function of the interpolation projection operators for the qubits:

$$\mathbf{\Pi}_k = \mathbf{I}_2^{\otimes(2^n-1-k)} \otimes \mathbf{\Pi} \otimes \mathbf{I}_2^{\otimes k} \qquad , \qquad k \in \{0, 2^n - 1\} \tag{46}$$

leading for a 4-qubit system to the operator:

$$\mathfrak{S}_{n=4} = e^{-\frac{2i\pi}{16}(8\mathbf{\Pi}_3+4\mathbf{\Pi}_2+2\mathbf{\Pi}_1+\mathbf{\Pi}_0)} \tag{47}$$

Here one can make a parallel with angular momentum observables which are generators for rotations. The eigenvalues $\hbar m$ of the z component angular momentum $\boldsymbol{J}_z$ obey the following rules $-j \le m \le +j$ and the difference between successive values is $\Delta m = 1$. $j \ge 0$ is integer or half-integer and $\hbar^2 j(j+1)$ is the eigenvalue of the associated observable $\boldsymbol{J}^2 = \boldsymbol{J}_x^2 + \boldsymbol{J}_y^2 + \boldsymbol{J}_z^2$. For cardinality 2^n one has $j^{[n]} = \frac{2^n-1}{2}$ which is half-integer, the considered numerical system becomes:

$$\begin{aligned} m^{[n]} &\in \frac{1}{2}\{-2^n+1, ..., -1, +1, ..., +2^n-1\} \\ m^{[n]} + j^{[n]} = d^{[n]} &\in \{0, 1, 2..., +2^n-1\} \end{aligned} \tag{48}$$

this shows that by shifting the spectrum one can express $\mathfrak{S}$ as a function of $\boldsymbol{J}_z$ leading to the operator expression:

$$\mathfrak{S} = -e^{+\frac{i\pi}{2^n}}e^{-\frac{2i\pi}{2^n}\frac{\boldsymbol{J}_z}{\hbar}} \tag{49}$$

The operator $\mathfrak{S}$ which is a function of the physical observable angular momentum can be thus associated with physical systems. The operator $\boldsymbol{U}_{FT}$ according to (45) can be expressed as a function of the product of two angular momentum operators $\boldsymbol{J}_z$ which could represent magnetic spin-spin interaction Hamiltonians.

4.2 $\{+1, 0, -1\}$ OAM system, for ternary Min and Max logical gates

The logical system $\{+1, 0, -1\}$ has several benefits because it approaches the two logical systems most commonly used in binary logic $\{+1, -1\}$ and $\{0, 1\}$, which are special cases of the considered ternary logic. Moreover, its values are centered in zero, thus assuring a simplification of the results and interesting properties of symmetry.

Orbital angular momentum (OAM) is characterized by two quantum numbers: ℓ the orbital number and m the magnetic number. The rules are: $\ell \geq 0$ is an integer and $-\ell \leq m \leq \ell$. The matrix form of the z-component orbital angular momentum observable for $\ell = 1$ is:

$$\boldsymbol{L}_z = \hbar \boldsymbol{\Lambda} = \hbar \begin{pmatrix} +1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{50}$$

the three eigenvalues $\{+1, 0, -1\}$ are here considered the logical truth values.

One can now express the ternary logical observables as developments over the rank-1 projection operators spanning the vector space: $\mathbf{\Pi}_{+1}$, $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_{-1}$. These operators are explicitly calculated using equation (1) and correspond to the density matrices of the three eigenstates $|+1>$, $|0>$ and $|-1>$ of $\boldsymbol{L}_z$, defining a qutrit. The projection operators function of the seed operator $\boldsymbol{\Lambda}$ are given by:

$$\mathbf{\Pi}_{+1} = \frac{1}{2}\boldsymbol{\Lambda}\left(\boldsymbol{\Lambda} + \boldsymbol{I}\right) \quad , \quad \mathbf{\Pi}_0 = \boldsymbol{I} - \boldsymbol{\Lambda}^2 \quad , \quad \mathbf{\Pi}_{-1} = \frac{1}{2}\boldsymbol{\Lambda}\left(\boldsymbol{\Lambda} - \boldsymbol{I}\right) \tag{51}$$

All arity-one logical operators $\boldsymbol{F}(\boldsymbol{\Lambda})$ can then be derived using the development (3).

When considering an arity-2 , 3-valued system, the operators are represented by 9×9 matrices. The dictators, $\boldsymbol{U}$ and $\boldsymbol{V}$, are then:

$$\boldsymbol{U} = \boldsymbol{\Lambda} \otimes \boldsymbol{I} \quad , \quad \boldsymbol{V} = \boldsymbol{I} \otimes \boldsymbol{\Lambda} \quad , \quad \boldsymbol{U} \cdot \boldsymbol{V} = \boldsymbol{\Lambda} \otimes \boldsymbol{\Lambda} \tag{52}$$

In ternary logic, popular connectives are Min and Max, defined by their truth-value maps in Table 1.

Using (51) and (52) and logical reduction rules (due to the completeness of the projection operators) one obtains the following observables:

$$\begin{cases} \mathbf{Min}(\boldsymbol{U}, \boldsymbol{V}) = \frac{1}{2}\left(\boldsymbol{U} + \boldsymbol{V} + \boldsymbol{U}^2 + \boldsymbol{V}^2 - \boldsymbol{U} \cdot \boldsymbol{V} - \boldsymbol{U}^2 \cdot \boldsymbol{V}^2\right) = diag(+1, +1+1, +1, 0, 0, +1, 0, -1) \\ \\ \mathbf{Max}(\boldsymbol{U}, \boldsymbol{V}) = \frac{1}{2}\left(\boldsymbol{U} + \boldsymbol{V} - \boldsymbol{U}^2 - \boldsymbol{V}^2 + \boldsymbol{U} \cdot \boldsymbol{V} + \boldsymbol{U}^2 \cdot \boldsymbol{V}^2\right) = diag(+1, 0, -1, 0, 0, -1, -1, -1, -1) \end{cases} \tag{53}$$

Min : $U \setminus V$	$+1$	0	-1
$+1$	$+1$	$+1$	$+1$
0	$+1$	0	0
-1	$+1$	0	-1

Max : $U \setminus V$	$+1$	0	-1
$+1$	$+1$	0	-1
0	0	0	-1
-1	-1	-1	-1

Table 1: The Min and Max tables for a three-valued two-argument logic

4.3 Operators for a qutrit balanced half-adder $\{-1, 0, +1\}$

The adder is one of the fundamental elements in digital electronics. One can, by the means of multivalued logic, improve the performances of this circuit by removing the delays caused by the propagation of the carry bit.

Considering a balanced half-adder, the structure is simplified because the ternary logic values match the notation for balanced ternary digits. The truth table for a balanced ternary half adder is given in Table 2 [20], where $\mathrm{S_i}$ represents the sum and $\mathrm{C_{i+1}}$ the carry-out logical connectives. The inputs for the half-adder are the addend value A and the carry-in value $\mathrm{C_i}$.

$\mathrm{S_i : A \setminus C_i}$	-1	0	$+1$
-1	$+1$	-1	$+0$
0	-1	0	$+1$
$+1$	0	$+1$	-1

$\mathrm{C_{i+1} : A \setminus C_i}$	-1	0	$+1$
-1	-1	0	0
0	0	0	0
$+1$	0	0	$+1$

Table 2: The Half-Adder tables for a balanced three-valued two-argument logic

The seed operator is reversed from the one given above: $\boldsymbol{\Lambda}_{ha} = diag(-1, 0, +1)$. The inputs are represented by the addend operator $\boldsymbol{A} = \boldsymbol{\Lambda}_{ha} \otimes \boldsymbol{I}$ and the carry-in operator $\boldsymbol{C}_i = \boldsymbol{I} \otimes \boldsymbol{\Lambda}_{ha}$. The output operators are the sum $\boldsymbol{S}_i$ and the carry-out $\boldsymbol{C}_{i+1}$ operators, their spectrum correspond to the respective truth values and is explicitly given in (54). The corresponding logical observables are derived directly using the truth tables and the respective projection operators (same method as for the OAM case using (51)):

$$\begin{cases} \boldsymbol{S}_i(\boldsymbol{A}, \boldsymbol{C}_i) \;=\; \boldsymbol{A} + \boldsymbol{C}_i - \dfrac{3}{2}\boldsymbol{A}^2 \cdot \boldsymbol{C}_i - \dfrac{3}{2}\boldsymbol{A} \cdot \boldsymbol{C}_i^2 = diag(+1, -1, 0, -1, 0, +1, 0, +1, -1) \\ \\ \boldsymbol{C}_{i+1}(\boldsymbol{A}, \boldsymbol{C}_i) \;=\; \dfrac{1}{2}\left(\boldsymbol{A} \cdot \boldsymbol{C}_i^2 + \boldsymbol{A}^2 \cdot \boldsymbol{C}_i\right) = diag(-1, 0, 0, 0, 0, 0, 0, 0, +1) \end{cases} \tag{54}$$

The comparison of the truth tables using ternary logic with the binary ones shows that the sum function $\mathrm{S_i}$ corresponds here to the modulo-3 sum function, and the carry function $\mathrm{C_{i+1}}$ is the consensus function. Therefore, the implementation of a balanced ternary half-adder is natural. The half-adder will either increment or decrement, depending on whether $\mathrm{C_0}$ is $+1$ or -1 , while the binary equivalent can only increment.

5 Conclusion

A general method has been presented for the design of logical quantum gates. It uses matrix interpolation inspired from classical multivariate methods. The interesting property is that a unique seed operator generates the entire logical family of operators for a given m-valued n-arity system. When considering the binary alphabet $\{+1, -1\}$ the gates are the quantum equivalent of the Fourier transform of Boolean functions. The method proposes a new expression of the Toffoli gate which can be understood as a 3-argument conjunction in the Eigenlogic interpretation. For multivalued logic quantum gates can be derived for different alphabets. The correspondence with quantum angular momentum leads to physical realizable gates. Applications have been presented for quantum Fourier transform gates, Min-Max and half-adder logical operations.

This opens a new perspective for quantum computation because several of the Eigenlogic operators [2, 3] turn out to be well-known quantum gates. This shows an operational correspondence between quantum control logic (Deutsch's paradigm [1]) and ordinary propositional logic. In Eigenlogic measurements on logical operators give the truth values of the corresponding logical connective. How could these measurement be exploited in a quantum circuit? Quantum tomography inspired techniques could be used.

At first sight the methods discussed here could be viewed as "classical" because of the identification of Eigenlogic with propositional logic. But when considering quantum states, that are not eigenvectors, the measurement outcomes are governed by the probabilistic quantum Born rule, and interpretable results are then the mean values, this leads to a fuzzy logic interpretation as outlined in [3].

Acknowledgements

We would like to thank Benoît Valiron from CentraleSupélec and LRI (Laboratoire de Recherche en Informatique), Gif-sur-Yvette (FR) for fruitful discussions during an ongoing academic project on quantum programming and for having pointed out the work using T gates. We are also very grateful to Francesco Galofaro, from Politecnico Milano (IT) and Free University of Bolzano (IT) for his pertinent advices on semantics and logic.

We much appreciated the feedback from the *Quantum Interaction* community these last years, the foundational aspects related to this work were presented at the QI-2016 conference in San Francisco and we wish to thank particularly Acacio da Barros of San Francisco State University (CA, ,USA), Ehtibar Dzhafarov of Purdue University (IN, USA), Andrei Khrennikov of Linnaeus University (SWE), Dominic Widdows from Microsoft Bing Bellevue (WA, USA), Peter Wittek from ICFO (Barcelona ESP) and Keith van Rijsbergen from University of Glasgow (UK).

References

[1] David Deutsch, “Quantum theory, the Church-Turing principle and the universal quantum computer”, Proc. Royal Soc. A. 400 (1818): pp. 97–117, 1985.

[2] Zeno Toffano, “Eigenlogic in the spirit of George Boole”, ArXiv:1512.06632, 2015.

[3] François Dubois and Zeno Toffano, “Eigenlogic: a Quantum View for Multiple-Valued and Fuzzy Systems”, Quantum Interaction. QI 2016. Lecture Notes in Computer Science, vol 10106. Springer, pp. 239-251, 2017, arXiv:1607.03509 [quant-ph].

[4] Pierre Cartier, “A mad day’s work: from Grothendieck to Connes and Kontsevich The evolution of concepts of space and symmetry”, Journal: Bull. Amer. Math. Soc. 38 (2001), 389-408.

[5] John von Neumann, “Mathematische Grundlagen der Quantenmechanik”, in “Grundlehren der mathematischen Wissenschaften”, volume Bd. 38. (Springer, Berlin, 1932) 106. “Mathematical Foundations of Quantum Mechanics”, Investigations in Physics, vol. 2, Princeton University Press, Princeton, 1955.

[6] Garret Birkhoff, John von Neumann, “The Logic of Quantum Mechanics”. The Annals of Mathematics, 2nd Ser., 37 (4), 823-843, 1936.

[7] Jeffrey Bub, “Quantum computation from a quantum logical perspective”, arXiv:quant-ph/0605243.

[8] Mingsheng Ying, “A theory of computation based on quantum logic (I)”, Theoretical Computer Science, Volume 344, Issues 2–3, Pages 134-207, 2005.

[9] Emil Post, “Introduction to a General theory of Elementary Propositions ”, American Journal of Mathematics 43: 163–185, 1921.

[10] Svetlana N. Yanushkevich, Vlad P. Shmerko, “Introduction to Logic Design”, CRC Press, 2008.

[11] Noboru Kikuchi, “Finite Element Methods in Mechanics”, Cambridge University Press, 1986.

[12] David Mermin, “Quantum Computer Science. An Introduction”, Cambridge University Press, 2007.

[13] George Boole, “The Mathematical Analysis of Logic. Being an Essay To a Calculus of Deductive Reasoning”, reissued Ed. Forgotten Books ISBN 978-1444006642-9, 1847.

[14] Ryan O’Donnell, “Analysis of Boolean Functions”, Cambridge University Press, 2014.

[15] Ashley Montanaro, Tobias J. Osborne, “Quantum boolean functions”, arXiv:0810.2435 [quant-ph]

[16] D. E. Muller, “Application of Boolean Algebra to Switching Circuit Design and to Error Detection”, IRE Transactions on Electronic Computers, 3:6–12, 1954.

[17] Matthew Amy, Michele Mosca, “T-count optimization and Reed-Muller codes”, arXiv:1601.07363.

[18] Peter Selinger, “Quantum circuits of T-depth one”, Phys. Rev. A 87, 042302, 2013.

[19] D. Michael Miller, Mitchell Aaron Thornton, “Multiple Valued Logic: Concepts and Representations”, Morgan & Claypool Publishers, 2008.

[20] Douglas W. Jones, “Fast Ternary Addition”, http://www.cs.uiowa.edu/~jones/ternary/ , University of Iowa Dep. of Computer Science, 2013.