4-22-2009

Quantum Multiplexers, Parrondo Games, and Proper Quantization

Faisal Shah Khan
Portland State University

QUANTUM MULTIPLEXERS, PARRONDO GAMES, AND PROPER QUANTIZATION

by

FAISAL SHAH KHAN

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY
in
MATHEMATICAL SCIENCES

Portland State University
2009

DISSERTATION APPROVAL

The abstract and dissertation of Faisal Shah Khan for the Doctor of Philosophy in Mathematical Sciences were presented April 22, 2009 and accepted by the dissertation committee and the doctoral program.

COMMITTEE APPROVALS: [redacted]
Steven Bleiler, Chair

[redacted]
Bin Jiang

[redacted]
Gerardo Lafferriere

[redacted]
Marek Perkowski

[redacted]
Bryant York
Representative of the Office of Graduate Studies

DOCTORAL PROGRAM APPROVAL: [redacted]
Steven Bleiler, Director
Mathematical Sciences Ph.D. Program

ABSTRACT

An abstract of the dissertation of Faisal Shah Khan for the Doctor of Philosophy in Mathematical Sciences presented April 22, 2009.

Title: Quantum Multiplexers, Parrondo Games, and Proper Quantization

A quantum logic gate of particular interest to both electrical engineers and game theorists is the quantum multiplexer. This shared interest is due to the facts that an arbitrary quantum logic gate may be expressed, up to arbitrary accuracy, via a circuit consisting entirely of variations of the quantum multiplexer, and that certain one player games, the history dependent Parrondo games, can be quantized as games via a particular variation of the quantum multiplexer. However, to date all such quantizations have lacked a certain fundamental game theoretic property.

The main result in this dissertation is the development of quantizations of history dependent quantum Parrondo games that satisfy this fundamental game theoretic property. Our approach also yeilds fresh insight as to what should be considered as the proper quantum analogue of a classical Markov process and gives the first game theoretic measures of multiplexer behavior.

This dissertation is dedicated to the following individuals: foremost, to my wife Seema and to my sons Arsalaan and Armaan for exhibiting a remarkable sense of humor toward my "changed" state of being during the days I wrote up this document. To my late father Haroon Shah Khan whose words "study finance, you clearly cannot do math!" inspired me to embark on this mathematical journey in the first place. To my brother Farrukh Shah Khan whose habitual philosophical ramblings and his ability to inspire through truly fantastical science fictional ideas and stories (not to mention financial support during my undergraduate studies) gave me the intellectual fortitude to reach this ultimate stage in my student career. And finally, to my mother Safia Bano and my sisters Farah Haroon, Fakhra Haroon, and Fadia Haroon, for putting up with me.

ACKNOWLEDGEMENTS

I thank Professors Marek Perkowski and Steven Bleiler for their guidance and support without which this work would not have been possible. This dissertation was composed with LaTex using a modified version of the thesis template created by Mark Paskin.

TABLE OF CONTENTS

Acknowledgements **ii**

List of Figures **v**

List of Tables **vii**

1 Quantum Mechanics and Computation **1**

1.1 Introduction to Quantum Computation 3

1.2 Quantum Computing with Multiple Qubits 12

2 A Formal Approach to Quantum Games **21**

2.1 The Bleiler Formalism for Quantum Mixtures 23

3 Properly Quantizing History Dependent Parrondo Games **32**

3.1 Parrondo Games 32

3.2 The FNA Quantization of Parrondo Games 45

3.3 Properly Quantizing History Dependent Parrondo Games 52

3.4 Properly Quantizing Randomized Sequences of History Dependent Parrondo Games 57

3.5 A Second Proper Quantization of the Randomized Sequence of History Dependent Parrondo Games 64

4 Quantum Logic Synthesis by Decomposition 68

4.1 The Cosine-Sine Decomposition (CSD) 70

4.2 Synthesis of 2-valued (binary) Quantum Logic Circuits 71

4.3 CSD Synthesis of 3-valued (ternary) Quantum Logic Circuits 75

4.4 Synthesis of Hybrid and d-valued Quantum Logic Circuits 78

5 A Quaternionic Co-ordinatization of Binary Quantum Computation 90

5.1 Identifying $SU(2)$ with $Sp(1)$ 91

5.2 Identifying $Sp(1)$ with $\mathbb{C}P^1$ 95

6 Future Directions 99

References 101

Appendix A Quaternions 104

Appendix B Cosine Sine Decomposition of Unitary Matrices 112

Appendix C List of Notations and Nomenclature 123

LIST OF FIGURES

1.1 Inverter or the NOT quantum logic gate U. The wires carry quantum information, namely qubits. 8

1.2 Standard notation for the NOT gate. 8

1.3 The Hadamard gate H that puts a basis state into an equal superposition of the basis states, and an arbitrary state $|\alpha\rangle$ into the superposition $\alpha_0 |0\rangle + \alpha_1 |1\rangle$. . 10

1.4 The two qubit Hadamard gate is just the tensor product of the one qubit Hadamard gates acting on each qubit. In general, multiqubits gates can be created via the tensor product of one qubit gates. However, it is not always true that a multiqubit gate is equal to the tensor product of one qubit gates. Consider for example the CNOT gate of Figure 1.5. 14

1.5 The controlled NOT (CNOT) gate. The vectors $|00\rangle$ and $|01\rangle$ are unaltered, while the vector $|10\rangle$ is sent to $|11\rangle$ and vice versa. 16

1.6 The Hadamard gate H that puts a basis state into an equal superposition of the basis states, and an arbitrary state $|\alpha\rangle$ into the superposition $\alpha_0 |0\rangle + \alpha_1 |1\rangle$. . 17

2.1 Extension of the game G to G^{mix}. 26

2.2 Extension of the game G to G^{Θ}. 29

2.3 Extension of the game G^{Θ} to $G^{m\Theta}$. 30

2.4 Proper quantization of a one player game with strategy space S via the protocol Θ and quantum strategy space Q. 30

3.1 Part of the quantization protocol for the history dependent Parrondo game. The first two wires represent the history qubits. 47

3.2 Proper Quantization, using the embedding e, of the History Dependent Game via the quantization protocol Θ_I. 56

4.1 2-valued Quantum Multiplexer M controlling the lower $(n-1)$ qubits by the top qubit. The slash symbol (/) represents $(n-1)$ qubits on the second wire. The gates labeled +1 are shifters (inverters in 2-valued logic), increasing the value of the qubit by 1 mod 2 thereby allowing for control by the highest qubit value. Depending on the value of the top qubit, one of U_t is applied to the lower qubits for $t \in \{0,1\}$. 72

4.2 A uniformly $(n-1)$-controlled R_y rotation for 2-valued quantum logic. The ○ control turns on for control value $|0\rangle$ and the • control turns on for control value $|1\rangle$. It requires 2^{n-1} one qubit controlled gates $R_y^{\theta_i}$ to implement a uniformly $(n-1)$-controlled R_y rotation. 73

4.3 A control by input value 0 (mod 2) realized in terms of control by the highest value 1 (mod 2). 74

4.4 A uniformly 2-controlled R_y rotation in 2-valued logic: the lower two qubits are the control qubits and the top bit is the target bit. 74

4.5 3-valued Quantum Multiplexer M controlling the lower $(n-1)$ qutrits via the top qutrit. The slash symbol (/) represents $(n-1)$ qutrits on the second wire. The gates labeled +2 are shift gates, increasing the value of the qutrit by 2 mod 3, and the control $\diamondsuit$ turns on for input $|2\rangle$. Depending on the value of the top qutrit, one of U_t is applied to the lower qutrits for $t \in \{0,1,2\}$. 76

4.6 A control by the value 0 (mod 3) realized in terms of control by the highest value 2 (mod 3). 76

4.7 A control by the value 1 (mod 3) realized in terms of control by the highest value 2 (mod 3). 77

4.8 A uniformly $(n-1)$-controlled R_x rotation. The lower $(n-1)$ qutrits are the control qutrits. The controls ○, •, and ◇ turn on for inputs $|0\rangle$, $|1\rangle$, and $|2\rangle$ respectively. It requires 3^{n-1} one qutrit controlled gates to implement a uniformly $(n-1)$-controlled R_x or R_z rotation. 77

4.9 An n qudit hybrid quantum multiplexer, here realized in terms of Muthukrishnan-Stroud (d-valued controlled) gates. The top qudit has dimension d_i and controls the remaining $(n-1)$ qudits of possibly distinct dimensions which are represented here by the symbol (/). The control $\oslash$ turns on for input value $|d_i - 1\rangle$ mod d_i of the controlling signal coming from the top qudit.The gates $+(d_i-1)$ shift the values of control qudit by (d_i-1) mod d_i. 86

4.10 A hybrid uniformly $(n-1)$-controlled Givens rotation. The lower $(n-1)$ qudits of dimensions $d_2, d_3, \ldots, d_{i-1}, d_{i+1}, \ldots, d_n$, respectively, are the control qudits, and the top is the target qudit of dimension d_i. The control gate $d_l^{(k)}$ turns on whenever the control qudit of dimension d_l takes on the value k (mod) d_l. 87

LIST OF TABLES

2.1 Prisoner's Dilemma 25
2.2 Simplified Poker. 25

3.1 Game B 34
3.2 History dependent game B'. 37

Chapter 1

QUANTUM MECHANICS AND COMPUTATION

Advances in computation technology over the last two decades have roughly followed Moore's Law, which asserts that the number of transistors on a microprocessor doubles approximately every two years. Extrapolating this trend, somewhere between the years of 2020 and 2030 circuits on a microprocessor will measure on an atomic scale. At this scale, quantum mechanical effects will materialize, and virtually every aspect of microprocessor design and engineering will be required to account for these effects.

To this end, *quantum information theory* studies information processing under a quantum mechanical model. One goal of the theory is the development of quantum computers with the potential to harness quantum mechanical effects for superior computational capability. In addition, attention will have to be paid to quantum mechanical effects that may obstruct coherent computation.

The study of possible development of quantum computers falls under the theory of quantum computation, an implementation of quantum information theory. Quantum computation model quantum information units, called *qudits,* as elements of a projective d-dimensional complex Hilbert space. Physical operations on the qudits are represented by unitary matrices and are viewed as quantum logic circuits. Major results in quantum computing demonstrate properties of quantum information that are not endemic to classical information. Contemporary data implies that in various aspects, quantized information offers advantages over classical information. For example, the Deutsch-Jozsa quantum algorithm [9] determines whether a function of n binary variables has a specific property or not in only one evaluation of the function, compared to the $2^{n-1}+1$ evaluations required by the deterministic non-quantum algorithm. Similarly, Grover's quantum search algorithm [13] searches a list in time that is quadratic rather than exponential in the number of elements in the list, and Shor's period finding quantum algorithm [32] gives a polynomial time algorithm for factoring integers. The last two are well known results in quantum computation and they show that quantum algorithms have the potential to out perform classical algorithms for practical problems.

Quantum game theory offers an exciting and relatively new game theoretic perspective on quantum information. Typically, research in the subject looks for different than usual behavior of the payoff function of a game when the game is played in a quantum mechanical setting. In multi-player games played in a quantum mechanical setting, the different than usual behavior of the payoff function studied is typically the occurrence of Nash equilibria that are absent in the original game [10, 18, 19]. Because quantum game theory has traditionally been heuristic in nature, confusion about and controversy over the relevance of "quantum games" to game theory abounds. A resolution to this confusion and controversy has been recently proposed by Bleiler in [5] via a mathematically formal approach to quantum game theory.

Using Bleiler's mathematically formal approach to quantum game theory as a stepping stone, this dissertation promotes the philosophy that quantum game theory should be used to gain insights into quantum computation. To this end, the reader is provided with a basic introduction to quantum computation and quantum mechanics in the remaining sections of this chapter. In Chapter 2, the Bleiler formalism is reproduced to give readers a mathematically formal game-theoretic perspective on quantum games. Chapter 3 presents the main results, which are construction and and game theoretic analysis of quantum versions of certain one player games, known as history dependent Parrondo games, and their randomized sequences using the Bleiler formalism as a blueprint. These constructions utilize a particular version of a quantum logic circuit known as the quantum multiplexer. The connection between quantum game theory and quantum computation is made apparent in Chapter 4, where the importance of the quantum multiplexer to quantum computation is established via abstract realization of an arbitrary quantum logic circuit in terms of circuits composed entirely of quantum multiplexers. Chapter 5 may be treated as a stand alone chapter; it proposes the analysis of quantum circuits acting on exactly two quantum informational units (qubits) via quaternionic coordinates.

1.1 Introduction to Quantum Computation

Like geometry, quantum mechanics is best viewed axiomatically. For the axioms of and basic facts about quantum mechanics, the reader is referred to [26, 27, 6]. These axioms and some of the basic facts appear explicitly in the next section during the development of one qubit quantum computation.

A d-ary quantum digit, or *qudit* for short, is a vector in a complex projective d-

dimensional Hilbert space $\mathcal{H}_d$, called the state space of the digit, equipped with the orthogonal *computational basis*

$$\{|0\rangle, |1\rangle, \ldots |d-1\rangle\}$$

where $|i\rangle = (0, 0, \ldots, 1, \ldots, 0)^T$ with a 1 in the $(i+1)$-st coordinate, for $0 \leq i \leq (d-1)$. To pass from classical to quantum computing, replace a classical d-ary digit (dit) with a qudit as an information unit. The replacement amounts to identifying all possible values of the dit with the elements of the computational basis of the state space of the corresponding qudit. This identification enlarges the set of operations on the dit to include quantum operations which, by the axioms of quantum mechanics, are represented by unitary operators on the state space. One then typically explores whether this enlargement results in any computational advantages or enhancements.

To be more specific, unitary operators can be used to create complex projective linear combinations of the basis qudits. In other words, a qudit $|a\rangle$ in $\mathcal{H}_d$ can be expressed as a complex projective linear combination of the basis qudits

$$|a\rangle = \sum_{i=0}^{d-1} x_i \, |i\rangle, \quad x_i \in \mathbb{C}$$

where $|a\rangle \equiv \lambda \, |a\rangle$ for any non-zero complex number λ. Physicists call this complex number λ a phase. Up to phase, the state $|a\rangle$ can be normalized; that is, $|a\rangle$ can be expresses with

$$\sum_{i=0}^{d-1} |x_i|^2 = 1$$

The measurement axioms of quantum mechanics say that the real number $|x_i|^2$ is the probability that the state vector $|a\rangle$ will be observed in i-th basis state upon measurement with respect to that basis. Typical considerations in quantum computing are whether evolutions of the state space offer computational enhancements.

When considering several qudits at once, the axioms of quantum mechanics tell us to consider their joint state space. When the state spaces of n qudits of different d-valued dimensions are combined, they do so via their tensor product as per the axioms of quantum mechanics and the result is a n qudit *hybrid* state space

$$\mathcal{H} = \mathcal{H}_{d_1} \otimes \mathcal{H}_{d_2} \otimes \cdots \otimes \mathcal{H}_{d_n}$$

where $\mathcal{H}_{d_i}$ is the state space of the d_i-valued qudit. The computational basis for $\mathcal{H}$ consists of all possible tensor products of the computational basis vectors of the component state spaces $\mathcal{H}_{d_i}$. If $d_i = d$ for each i, the resulting state space $\mathcal{H}_d^{\otimes n}$ is that of n d-valued qudits.

Once a basis for the state space has been chosen, a unitary operator on it is represented by a unitary matrix. For the hybrid state space $\mathcal{H}$, an evolution matrix will be of size $(d_1 d_2 \dots d_N) \times (d_1 d_2 \dots d_N)$, while the evolution matrix for $\mathcal{H}_d^{\otimes n}$ will have size $d^n \times d^n$.

Consider a two dimensional state space $\mathcal{H}_2$. This is the state space of a quantum system which gives two possible outcomes upon measurement. An example of such a system would be one that describes the spin states of an electron. Topologically, $\mathcal{H}_2 = \mathbb{C}P^1$. The two possible states of the system form the computational basis for the state space. These orthogonal basis state are viewed as the two possible values a bit of information can take on. Call the elements of $\mathcal{H}_2$ *qubits*, short for binary quantum digit. The resulting 2-valued quantum computing has traditionally been the most active area of research. The basics of 2-valued quantum computing are reviewed in the following sections. Higher valued quantum computing has seen much research activity recently as well. The reader is referred to chapter 2 for a discussion of certain aspects of d-valued quantum computing and relevant references.

1.1.1 One Qubit Quantum Computing

Let $|b_0\rangle$ and $|b_1\rangle$ be an orthogonal basis for $\mathcal{H}_2$. Then the states of the qubit are projective linear combinations of these basis elements over $\mathbb{C}$:

$$|\psi\rangle = \alpha_0 \, |b_0\rangle + \alpha_1 \, |b_1\rangle$$

with $\alpha_0, \alpha_1 \in \mathbb{C}$ satisfying, without loss of generality, $|\alpha_0|^2 + |\alpha_1|^2 = 1$. These projective complex linear combinations are also called *superpositions* of the states $|0\rangle$ and $|1\rangle$. The computational basis is the set

$$\mathcal{B}\text{comp} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

which gives the convention of labeling the basis with Boolean names, with

$$|b_0\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |b_1\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

But note that these are only names. For example, in the spin state model for an electron, one might imagine that $|0\rangle$ is being represented by an up-spin while $|1\rangle$ by a down-spin. The key is that there is an abstraction between the technology (spin state or other quantum phenomena) and the logical meaning. This same detachment is true in classical computers where we traditionally call a high positive voltage “1” and a low ground potential “0”.

Let $|\psi_1\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ and

$$U = \begin{pmatrix} 0 & -\overline{\eta} \\ \eta & 0 \end{pmatrix} \tag{1.1}$$

be a special unitary operator which, by axioms of quantum mechanics, corresponds to a physical operation. Further, suppose that η is a complex root of unity other than ±1, the use of which will be justified shortly. The matrix U acts on $|\psi_1\rangle$ as follows.

$$U|\psi_1\rangle = \begin{pmatrix} 0 & -\overline{\eta} \\ \eta & 0 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} -\overline{\eta}\alpha_1 \\ \eta\alpha_0 \end{pmatrix} = -\overline{\eta}\alpha_1 |0\rangle + \eta\alpha_0 |1\rangle. \tag{1.2}$$

Up to multiplication by unitary phase, the operator U interchanges the coefficients of

Figure 1.1: Inverter or the NOT quantum logic gate U. The wires carry quantum information, namely qubits.

Figure 1.2: Standard notation for the NOT gate.

the basis states of $\mathbb{C}P^1$. In particular, U sends the state $|0\rangle$ to the state $\eta\,|1\rangle$

$$U\,|0\rangle = \begin{pmatrix} 0 & -\bar{\eta} \\ \eta & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \eta \end{pmatrix} = \eta\,|1\rangle$$

and the state $|1\rangle$ to the state $-\bar{\eta}\,|0\rangle$

$$U\,|1\rangle = \begin{pmatrix} 0 & -\bar{\eta} \\ \eta & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\bar{\eta} \\ 0 \end{pmatrix} = -\bar{\eta}\,|0\rangle\,.$$

This action of U is interpreted as that of a *quantum logic gate* that inverts, up to unitary phase, the logical values $|0\rangle$ and $|1\rangle$; that is, the gate U is a quantum version of the NOT gate in classical logic. This point of view allows one to view quantum mechanics as a theory of quantum computation. Standard notation for the NOT gate is given in Figure 1.2.

In the quantum theory of games, one frequently views a qubit as a "quantum coin" and hence the gate U can be interpreted as the quantum mechanical analog of flipping over a coin, while the 2×2 identity matrix is the analog of leaving the coin un-flipped. In certain quantum games, such as the ones found in [18, 1], the flipping and un-flipping actions of players on the so-called maximally entangled state of two qubits

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

are considered. For the purpose of analysis of the quantum game, these actions are required to produce an orthogonal basis of the joint state space, and this happens only when η is an appropriate root of unity other than ± 1.

1.1.2 The One Qubit Hadamard Quantum Logic Gate

Quantum computing literature gives many interesting examples of one qubit gates. The focus here will be on the one qubit Hadamard gate described by the special unitary matrix

$$H = \frac{i}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Application of the gate H to either basis states $|0\rangle$ and $|1\rangle$ creates an *equal superposition* of the basis state, that is, a superposition that will appear in each basis state with equal

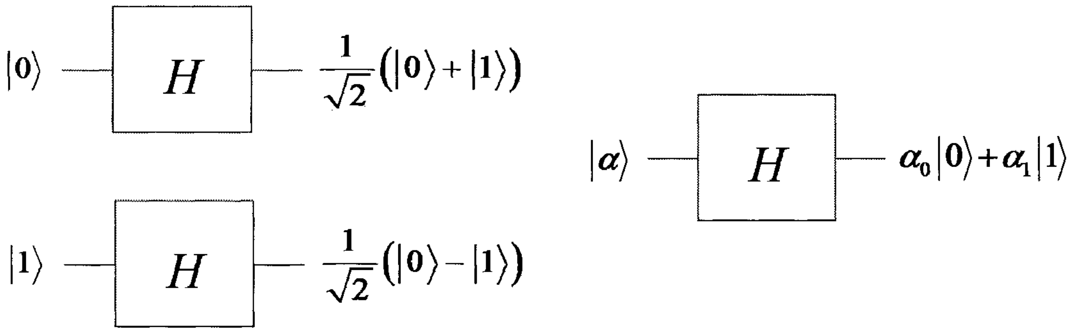

$|0\rangle$
H
$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
$|1\rangle$
H
$\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$
$|\alpha\rangle$
H
$\alpha_0|0\rangle+\alpha_1|1\rangle$

Figure 1.3: The Hadamard gate H that puts a basis state into an equal superposition of the basis states, and an arbitrary state $|\alpha\rangle$ into the superposition $\alpha_0 |0\rangle + \alpha_1 |1\rangle$.

probability upon measurement with respect to the basis.

$$\begin{aligned} H|0\rangle &= \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{i}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{i}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{i}{\sqrt{2}} |0\rangle + \frac{i}{\sqrt{2}} |1\rangle \end{aligned}$$

and

$$\begin{aligned} H|1\rangle &= \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \frac{i}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \frac{i}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{i}{\sqrt{2}} |0\rangle - \frac{i}{\sqrt{2}} |1\rangle . \end{aligned}$$

1.1.3 Measurement

In Equation (1.2), if both $\alpha_0, \alpha_1 \neq 0$, then the how does one interpret the complex projective linear combination $-\overline{\eta}\alpha_1 |0\rangle + \eta\alpha_0 |1\rangle$ of the basis states in the context of computing? The answer comes from quantum mechanics' axiom of measurement which allows a probabilistic interpretation of such complex projective linear combinations as follows. Upon measurement with respect to the orthogonal basis $\{|0\rangle, |1\rangle\}$, the combination is observed to be in the basis state $|0\rangle$ with probability $|\alpha_1|^2$ and in the basis state $|1\rangle$ with probability $|\alpha_1|^2$ (remember that η is a unit complex number so $|\eta|^2 = |\overline{\eta}|^2 = 1$). Two important measurement operators are

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

The measurement operator M_0 projects a complex projective linear combination onto the basis state $|0\rangle$ while M_1 projects onto the basis state $|1\rangle$. For example, let

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

Then the probability of measuring the complex projective linear combination $|\psi\rangle$ in the basis state $|0\rangle$ is

$$p(|0\rangle) = \begin{pmatrix} \overline{a} & \overline{b} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

$$= \begin{pmatrix} \overline{a} & \overline{b} \end{pmatrix} \begin{pmatrix} a \\ 0 \end{pmatrix} = |a|^2 .$$

Note that measurement operators are *not* quantum logic gates as they are non-unitary, but rather are projections onto the basis states.

1.2 Quantum Computing with Multiple Qubits

Quantum computing can be extended to multiple qubits via the creation of composite state spaces from the state spaces of many individual qubits.

For example, consider two qubits $|\psi_1\rangle = a\,|0\rangle + b\,|1\rangle$ and $|\psi_2\rangle = c\,|0\rangle + d\,|1\rangle$, both written with respect to the computational basis. Then the *joint state* of the total system is given by:

$$|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\psi_2\rangle = ac\,|0\rangle \otimes |0\rangle + ad\,|0\rangle \otimes |1\rangle + bc\,|1\rangle \otimes |0\rangle + bd\,|1\rangle \otimes |1\rangle$$

$$= ac\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} + ad\begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} + bc\begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix}$$
$$+bd\begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}$$

$$= ac\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + ad\begin{pmatrix}0\\1\\0\\0\end{pmatrix} + bc\begin{pmatrix}0\\0\\1\\0\end{pmatrix} + bd\begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

$$= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle$$

1.2.1 Two Qubit Quantum Gates

An easy way to obtain two qubit quantum gates is by producing the tensor product of two one qubit gates. That is, if U_1 and U_2 are one qubit gates, then

$$U = U_1 \otimes U_2$$

is a two qubit gate. Two qubit gates such as U above that are tensor products of one qubit gates act locally on each qubit due to the bi-linearity of the tensor product. Nonetheless,

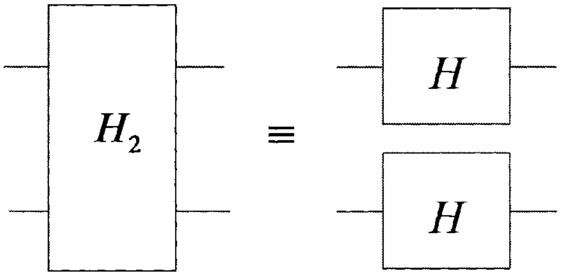


Figure 1.4: The two qubit Hadamard gate is just the tensor product of the one qubit Hadamard gates acting on each qubit. In general, multiqubits gates can be created via the tensor product of one qubit gates. However, it is not always true that a multiqubit gate is equal to the tensor product of one qubit gates. Consider for example the CNOT gate of Figure 1.5.

such gates are crucial to quantum computing. For example, the two qubit Hadamard gate defined as

$$H_2 = H \otimes H = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$= -\frac{1}{2} \begin{pmatrix} 1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & 1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ 1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & -1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{pmatrix} = -\frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

is essential for the creation of a particular equal superposition of two qubits which plays a crucial role in the development of quantum algorithms that out-perform classical algorithms [13, 32].

1.2.2 Controlled NOT (CNOT) gate

Perhaps the most important two qubit gate is the controlled NOT (CNOT) gate. Its importance lies in its property of forming, together with one qubit gates, sets of universal quantum logic gates. Informally, a set of quantum logic gates is *universal* if any quantum logic gate may be approximated by the gates in the set to arbitrary accuracy. For a detailed discussion of universality, the reader is refer

The CNOT gate acts as a NOT gate on the second qubit (target qubit) if the first qubit (control qubit) is in the computational basis state $|1\rangle$. So when passing through the gate the states $|00\rangle$ and $|01\rangle$ are unaltered, while the state $|10\rangle$is sent to $|11\rangle$ and vice versa. In the joint computational basis, the CNOT gate is

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Note that the CNOT gate is *not* the tensor product of any pair of one qubit gates. Indeed, there are plenty of other two and multiqubit gates that are not tensor products of one qubit gates. This property of quantum logic gates is one more reason that quantum logic circuit synthesis is a much studied subject.

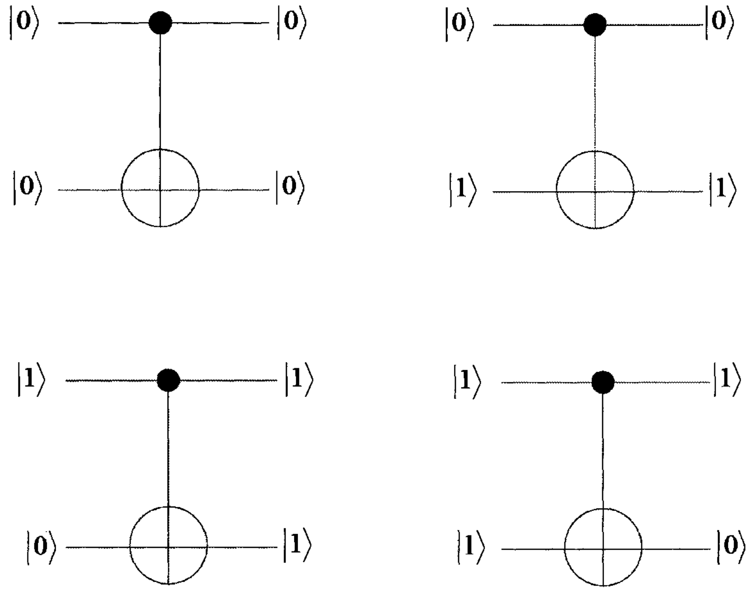


Figure 1.5: The controlled NOT (CNOT) gate. The vectors $|00\rangle$ and $|01\rangle$ are unaltered, while the vector $|10\rangle$ is sent to $|11\rangle$ and vice versa.

1.2.3 Entanglement

Entanglement is a uniquely quantum phenomenon. Entanglement is a property of a multi-qubit system and can be thought of as a resource. To explain entanglement, let us examine a so-called *EPR pair* of qubits named after Einstein, Podolsky, and Rosen. The CNOT gate will be used in this example.

We begin with two qubits $|\psi_1\rangle = |0\rangle$ and $|\psi_2\rangle = |1\rangle$. Apply the Hadamard gate to $|\psi_1\rangle$ to get

$$|\psi'{}_1\rangle = H\,|\psi_1\rangle = \frac{i}{\sqrt{2}}\,|0\rangle + \frac{i}{\sqrt{2}}\,|1\rangle$$

The joint state-space vector is the tensor product

$$|\psi'_1\rangle \otimes |\psi_2\rangle = |\psi'_1\psi_2\rangle = \frac{i}{\sqrt{2}}\,|00\rangle + (0)\,|01\rangle + \frac{i}{\sqrt{2}}\,|10\rangle + (0)\,|11\rangle$$

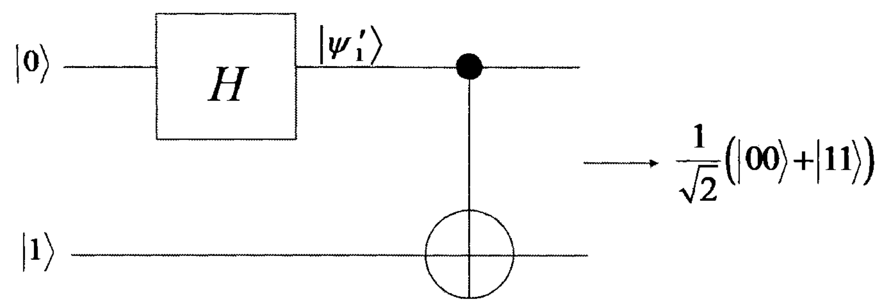


Figure 1.6: The Hadamard gate H that puts a basis state into an equal superposition of the basis states, and an arbitrary state $|\alpha\rangle$ into the superposition $\alpha_0\,|0\rangle + \alpha_1\,|1\rangle$.

Now apply the CNOT gate to this joint state of the two qubits. This gives

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{i}{\sqrt{2}} \\ 0 \\ \frac{i}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{i}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{i}{\sqrt{2}} \end{pmatrix} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{i}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \frac{i}{\sqrt{2}}\,|00\rangle + \frac{i}{\sqrt{2}}\,|11\rangle$$

The final joint state above has the property that it cannot be built up from the tensor product of states in the component spaces of each qubit. That is,

$$\frac{i}{\sqrt{2}}\,|00\rangle + \frac{i}{\sqrt{2}}\,|11\rangle \neq \ |\phi_1\rangle \otimes |\phi_2\rangle\,.$$

To illustrate why entanglement is so strange, let's consider performing a measurement just prior to applying the CNOT gate. The two measurement operators (for obtaining a $|00\rangle$or a$|11\rangle$) are:

$$M_{00} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } M_{11} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Just prior to the CNOT the system is in the state

$$\frac{i}{\sqrt{2}}|00\rangle + 0\,|01\rangle + \frac{i}{\sqrt{2}}|10\rangle + 0\,|11\rangle,$$

therefore

$$p(0) = \begin{pmatrix} \frac{\bar{i}}{\sqrt{2}} & 0 & \frac{\bar{i}}{\sqrt{2}} & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{i}{\sqrt{2}} \\ 0 \\ \frac{i}{\sqrt{2}} \\ 0 \end{pmatrix} = 1$$

Hence the result of measuring will clearly be $|0\rangle$. After the measurement, we have

$$|\psi_1'\psi_2\rangle = \frac{\begin{pmatrix} \frac{i}{\sqrt{2}} \\ 0 \\ \frac{i}{\sqrt{2}} \\ 0 \end{pmatrix}}{1}$$

and we see that measurement had no effect on the first qubit and it remains in a superposition of $|0\rangle$ and $|1\rangle$. Now consider the same measurement but just after the CNOT gate is applied, with the joint state $|\psi_3\rangle = \frac{i}{\sqrt{2}}|00\rangle + \frac{i}{\sqrt{2}}|11\rangle$.

$$p(0) == \begin{pmatrix} \frac{\bar{i}}{\sqrt{2}} & 0 & 0 & \frac{\bar{i}}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{i}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{i}{\sqrt{2}} \end{pmatrix} = \frac{1}{2}$$

Hence, after the *CNOT* gate is applied we have only a 50% chance of obtaining $|0\rangle$. Of particular interest to our discussion, however, is what happens to the state vector of the system after measurement.

$$\frac{\begin{pmatrix} \frac{i}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{pmatrix}}{\sqrt{\frac{1}{2}}} = i \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = i \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = i\,|00\rangle$$

This is the remarkable thing about entanglement. By measuring one qubit we can affect the probability of the state observations of the other qubits in a system! The state of the other qubit $|\psi_1'\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ is changed to $|0\rangle$ after the measurement.

Quoting Oskin [27] regarding entanglement:

"How to think about this process (entanglement) in an abstract way is an open challenge in quantum computing. The difficulty is the lack of any classical analog. One useful, but imprecise way to think about entanglement, superposition and measurement is that superposition "is" quantum information. Entanglement links that information across quantum bits, but does not create any more of it. Measurement "destroys" quantum information turning it into classical. Thus think of an EPR pair as having as much "superposition" as an un-entangled set of qubits, one in a superposition between zero and one, and another in a pure state. The superposition in the EPR pair is simply linked across qubits instead of being isolated in one."

Chapter 2

A FORMAL APPROACH TO QUANTUM GAMES

One way to view a game is as a function. We view here quantum games as extensions of such functions. For a detailed and formal introduction to game theory the reader is referred to [3] and [24]. The following discussion on quantum games that follows is motivated by a mathematical formalism for “quantum mixtures” developed by S. Bleiler in [5] and reproduced in section 2.1 below.

Recall that a key goal in the study of multi-player, non-cooperative games is the identification of potential Nash equilibria. Informally, a Nash equilibrium occurs when each player chooses to play a strategy that is a best reply to the choice of strategies of all the other players. In other words, unilateral deviation from the choice of strategy at a Nash equilibrium by any player is detrimental to that player’s payoff in the game. However, in finite classical games, Nash equilibria may not exist. In such situations, classical game theory calls upon the players to randomize between their strategic choices, also known as mixing strategies. For finite games, Nash proved [25] that this gives rise to Nash equilibria in the “mixed game” that simply do not exist in the original game. Formally, the mixed game is the result of an extension of the payoff function of the original game to a larger set of strategies for each player.

The Bleiler formalism for quantum mixtures views quantum game theory in this light. That is, this formalism views quantum game theory as an exercise in the extension of the payoff function of a game with the goal of finding Nash equilibria with higher payoffs that were un-attainable in the original game or its "classical extensions". The extensions dealt with in quantum game theory are referred to as a *quantization protocols*. This mathematically formal perspective provides a game theoretic context in which many issues in quantum game theory can be discussed and potentially resolved. For example, critics of quantum game theory wonder whether instances of Nash equilibira with higher payoffs in certain quantum games are just Nash equilibria of some other classical game theoretic construction realized quantum mechanically. This point of view implies that quantum game theory is essentially a study in expensive ways to generate classical game theoretic results and offers nothing "new" to game theory.

Such criticism is addressed in the Bleiler formalism which points out that any quantum game that contains the original or the classical game as an embedded subgame has the potential to offer something new to the game's analysis. When a quantum game has this property, it is referred to in the Bleiler formalism as a *proper quantization* of the original game. When a quantum game carries an embedded copy of the mixed version of the original game, the formalism refers to it as a *complete quantization* of the original game. Much of the current work in quantum game theory can be characterized as calling upon the players to use the higher orders of randomization given by quantum superpositions and randomized quantum superpositions. Call these *quantum strategies* and *mixed quantum strategies*, respectively. If the quantization of the game is proper or complete, then any new Nash equilibria with higher payoffs that result from the use of quantum or mixed quantum strategies can be meaningfully compared with the Nash equilibria of the original game.

A detailed review of the Bleiler formalism follows.

2.1 The Bleiler Formalism for Quantum Mixtures

Definition 2.1. Given a set $\{1, 2, \cdots, n\}$ of players, for each player a set S_i $(i = 1, \cdots, n)$ of so-called *pure strategies*, and a set Ω_i $(i = 1, \cdots, n)$ of *possible outcomes*, a *game* G is a vector-valued function whose domain is the Cartesian product of the S_i's and whose range is the Cartesian product of the Ω_i's. In symbols

$$G : \prod_{i=1}^{n} S_i \longrightarrow \prod_{i=1}^{n} \Omega_i$$

The function G is sometimes referred to as the *payoff function*.

Here a *play* of the game is a choice by each player of a particular strategy s_i the collection of which forms a *strategy profile* $(s_1, \cdots, s_n)$ whose corresponding *outcome profile* is $G(s_1, \cdots, s_n) = (\omega_1, \cdots, \omega_n)$, where the ω_i's represent each player's individual outcome. Note that by assigning a real valued *utility* to each player which quantifies that player's preferences over the various outcomes, we can without loss of generality, assume that the Ω_i's are all copies of $\mathbb{R}$, the field of real numbers.

In game theory, players' concern is the identification of a strategy that guarantees a maximal utility. For a fixed $(n-1)$-tuple of opponents' strategies, rational players seek a *best reply*, that is a strategy s^* that delivers a utility at least as great, if not greater, than any other strategy s. When every player can identify such a strategy, the resulting strategy profile is called a *Nash equilibrium.* Formally,

Definition 2.2. Let s_{-i} be a strategy profile of all players except player i. A *Nash equilibrium* (NE) for the game G is a strategy profile (s_i^*, s_{-i}) such that

$$G(s_i^*, s_{-i}) \geq G(s_i, s_{-i})$$

where for all i, $s_i, s_i^* \in S_i$ and $s_i^* \neq s_i$.

Other ways of expressing this concept include the observation that no player can increase his or her payoffs by unilaterally deviating from his or her equilibrium strategy, or that at equilibrium all of a player's opponents are indifferent to that player's strategic choice. As an example, consider the Prisoner's Dilemma, a two player game where each player has exactly two strategies (a so-called 2×2 or *bimatrix* game) and whose payoff function is indicated in Table 2.1. The rows of Table 2.1 contain the strategies of player 1 while the columns contain the strategies of player 2.

Note that for player 1 the pure strategy s_2 always delivers a higher outcome than the strategy s_1 (say s_2 *strongly dominates* s_1) and for player 2 the strategy t_2 strongly dominates t_1. Hence the pair (s_2, t_2) is a (unique) Nash Equilibrium.

However, games need not have equilibria amongst the pure strategy profiles as exemplified by the 2×2 game of Simplified Poker whose payoff function is given in Table 2.2.

As remarked above, the game theoretic formalism now calls upon the theorist to extend the game G by enlarging the domain and extending the payoff function. Of

	t_1	t_2
s_1	$(3,3)$	$(0,5)$
s_2	$(5,0)$	$(1,1)$

Table 2.1: Prisoner's Dilemma

	t_1	t_2
s_1	$(5/4,-5/4)$	$(0,0)$
s_2	$(0,0)$	$(5/2,-5/2)$

Table 2.2: Simplified Poker.

course, the question of if and how a given function extends is a time honored problem in mathematics and the careful application of the mathematics of extension is what will drive the formalism for quantization. Returning to classical game theory, a standard extension at this point is to consider for each player the set of mixed strategies.

Definition 2.3. A *mixed strategy* for player i is an element of the set of probability distributions over the set of pure strategies S_i.

For a given set X, denote the probability distributions over X by $\Delta(X)$ and note that when X is finite, with k elements say, the set $\Delta(X)$ is just the $k-1$ dimensional simplex $\Delta^{(k-1)}$ over X, i.e., the set of real convex linear combinations of elements of X. Of course, we can embed X into $\Delta(X)$ by considering the element x as mapped to the probability distribution which assigns 1 to x and 0 to everything else. For a given game G, denote this embedding of S_i into $\Delta(S_i)$ by e_i.

Let $p = (p_1, \ldots, p_n)$ be a mixed strategy profile. Then p induces the *product distribution* over the product $\prod S_i$. Taking the push out by G of the product distribution (i.e., given a probability distribution over strategy profiles, replace the profiles with their images under G) then gives a probability distribution over the image of G, ImG. Fol-

$$\begin{tikzcd}
\prod\Delta(S_i) \arrow[r, "\text{Product}"] \arrow[drr, "G^{mix}"] & \Delta(\prod S_i) \arrow[r, "\text{Push-out}"] & \Delta(\mathrm{Im}G) \arrow[d, "E"] \\
\prod S_i \arrow[u, dotted, "\prod e_i"] \arrow[rr, "G"] & & \prod\Omega_i
\end{tikzcd}$$

Figure 2.1: Extension of the game G to G^{mix}.

lowing this by the expectation operator E, we obtain the *expected outcome of* p. Now our game G can be extended to a new, larger game G^{mix}.

Definition 2.4. Assigning the expected outcome to each mixed strategy profile we obtain the extended game

$$G^{mix} : \prod \Delta(S_i) \rightarrow \prod \Omega_i$$

Note G^{mix} is a true extension of G as $G^{mix} \circ \Pi e_i = G$; that is, the diagram in Figure 2.1 is commutative.

As remarked above, Nash's famous theorem [25] says that if the S_i are all finite, then there always exists an equilibrium in G^{mix}. Unfortunately, this equilibrium is called a *mixed strategy equilibrium for* G, when it is not an equilibrium of G at all, the abusive terminology confusing G with its image, ImG.

2.1.1 Quantization

The Bleiler formalism asserts that some of the controversies surrounding quantum game theory may be resolved if one focuses on the quantization of the *payoffs* of the original game G, and expresses the quantized version of G as a (proper) extension of the original payout function in the set-theoretic sense, just as in the classical case.

Classically, probability distributions over the outcomes of a game G were constructed. Now the goal is to pass to a more general notion of randomization, that of quantum superposition. Begin then with a Hilbert space $\mathcal{H}$ that is a complex vector space equipped with an inner product. For the purpose here assume that $\mathcal{H}$ is finite dimensional, and that there exists a finite set X which is in one-to-one correspondence with an orthogonal basis $\mathcal{B}$ of $\mathcal{H}$. When the context is clear as to the basis to which the set X is identified, denote the set of quantum superpositions for X as $\mathcal{Q}S(X)$. Of course, it is also possible to define quantum superpositions for infinite sets, but for the purpose here, one need not be so general. What follows can be easily generalized to the infinite case.

As mentioned above, the underlying space of complex linear combinations is a Hilbert space; therefore, we can assign a length to each quantum superposition and, up to phase, always represent a given quantum superposition by another that has length 1.

For each quantum superposition of X we can obtain a probability distribution over X by assigning to each component the ratio of the square of the length of its coefficient to the square of the length of the combination. This assignment is in fact functional, and is abusively referred to as measurement. Formally:

Definition 2.5. *Quantum measurement with respect to* X is the function

$$q_X^{meas} : \mathcal{Q}S(X) \longrightarrow \Delta(X)$$

given by

$$\alpha x + \beta y \longmapsto \left(\frac{|\alpha|^2}{|\alpha|^2 + |\beta|^2}, \frac{|\beta|^2}{|\alpha|^2 + |\beta|^2} \right)$$

Note that geometrically, quantum measurement is defined by projecting a normalized quantum superposition onto the various elements of the normalized basis $\mathcal{B}$. Denote quantum measurement by q^{meas} if the set X is clear from the context.

Now given a finite n-player game G, suppose we have a collection $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$ of non-empty sets and a *protocol*, that is, a function $\Theta : \prod \mathcal{Q}_i \longrightarrow \mathcal{QS}(\mathrm{ImG})$. Quantum measurement q^{meas}_{ImG} then gives a probability distribution over ImG. Just as in the mixed strategy case we can then form a new game G^{Θ} by applying the expectation operator E.

Definition 2.6. Assigning the expected outcome to each probability distribution over ImG that results from quantum measurement, we obtain the quantized game

$$G^{\Theta} : \prod \mathcal{Q}_i \rightarrow \prod \Omega_i$$

Call the game G^{Θ} thus defined to be the *quantization of G by the protocol* Θ. Call the $\mathcal{Q}_i$'s sets of *pure quantum strategies* for G^{Θ}. Moreover, if there exist embeddings $e_i' : S_i \longrightarrow \mathcal{Q}_i$ such that $G^{\Theta} \circ \prod e_i' = G$, call G^{Θ} a *proper* quantization of G. If there exist embeddings $e_i'' : \Delta(S_i) \longrightarrow \mathcal{Q}_i$ such that $G^{\Theta} \circ \prod e_i'' = G^{mix}$, call G^{Θ} a *complete* quantization of G. These definitions are summed up in the commutative diagram of Figure 2.2. Note that for proper quantizations, the original game is obtained by restricting the quantization to the image of $\prod e_i'$. For general extensions, the Game Theory literature refers to this as "recovering" the game G.

It follows from the definitions of G^{mix} and G^{Θ} that a complete quantization is proper.

$$\begin{tikzcd}
 & & & & QS(\mathrm{Im}G) \arrow[dd, "q^{meas}_{\mathrm{Im}G}"] \\
 & & & \prod Q_i \arrow[ur, "\Theta"] \arrow[dd, "G^{\Theta}"] & \\
 & \prod \Delta(S_i) \arrow[r, "\text{Product}"] \arrow[urr, "\prod e''_i"] \arrow[drr, "G^{mix}"] & \Delta(\prod S_i) \arrow[rr, "\text{Push-out}"] & & \Delta(\mathrm{Im}G) \arrow[dl, "E"] \\
\prod S_i \arrow[ur, dashed, "\prod e_i"] \arrow[rrr, "G"] \arrow[uurrr, bend left, "\prod e'_i"] & & & \prod \Omega_i &
\end{tikzcd}$$

Figure 2.2: Extension of the game G to G^{Θ}.

Furthermore, note that finding a mathematically proper quantization of a game G is now just a typical problem of extending a function. It is also worth noting here that nothing prohibits us from having a quantized game G^{Θ} play the role of G in the classical situation and by considering the probability distributions over the Q_i, creating a yet larger game $G^{m\Theta}$, the *mixed quantization of G with respect to the protocol* Θ. For a proper quantization of G, $G^{m\Theta}$ is an even larger extension of G. The game $G^{m\Theta}$ is described in the commutative diagram of Figure 2.3.

In many cases, the Q_i of the quantization protocols are expressed as quantum operations. These operations require a state to "operate" on. In this situation the definition of protocol additionally requires the definition of an "initial state" together with the family of quantum operations which act upon this state, along with a specific definition of how these quantum operations are to act. As exemplified in the next chapter, different

$$\begin{tikzcd}
\prod\Delta(Q_i) \arrow[r, "\text{Product}"] \arrow[drr, "G^{mQ}"] & \Delta\left(\prod Q_i\right) \arrow[r, "\text{Push-out}"] & \Delta\left(\mathrm{Im}\, G^{Q}\right) \arrow[d, "E"] \\
\prod Q_i \arrow[u, dashed, "\prod \tilde{e}_i"] \arrow[rr, "G^{Q}"] & & \prod \Omega_i
\end{tikzcd}$$

Figure 2.3: Extension of the game G^{Θ} to $G^{m\Theta}$.

$$\begin{tikzcd}
(Q,I) \arrow[r, "\Theta_I"] \arrow[drr, "G_s^{\Theta_I}"] & QS(\mathrm{Im}\, G) \arrow[r, "q_{meas}^{\mathrm{Im}\, G}"] & \Delta(\mathrm{Im}\, G) \arrow[d, "E"] \\
(S,s) \arrow[u, dashed, "e"] \arrow[rr, "G_s"] & & \Omega
\end{tikzcd}$$

Figure 2.4: Proper quantization of a one player game with strategy space S via the protocol Θ and quantum strategy space Q.

choices for the initial state can give rise to very different protocols sharing a common selection and action of quantum operations. When a protocol Θ depends on a specific initial state I, the protocol is then denoted by Θ_I.

In subsequent sections, a version of this formalism adapted to one player games will be utilized. The underlying quantization paradigm being the replacement of probability distributions by the more general notion of quantum superposition followed by measurement. The functional diagram for proper quantization that will be utilized is given in Figure 2.4 where the commutativity of the diagram requires that $E \circ (q_{meas}^{\mathrm{Im}G}) \circ \Theta \circ e = G^{\Theta} \circ e = G$. Incorporating the discussion above, when games G_s and protocols Θ_I depend on a given initial states s and I, respectively, the initial states s and I are regarded as part of the single player's strategic choice. In these cases, the embedding e of S into Q additionally requires the mapping of the initial state s of G_s to the initial state I of the protocol Θ_I. The resulting quantum game is denoted by $G_s^{\Theta_I}$.

Chapter 3

PROPERLY QUANTIZING HISTORY DEPENDENT PARRONDO GAMES

A major insight about quantized games that results from the Bleiler formalism discussed in Chapter 2 is that for the quantization of a game to be game-theoretically significant, it must be proper. Previous work on the quantization of the history dependent Parrondo game by Flitney, Ng, and Abbott (FNA) [11] produced quantizations that are not proper. In this chapter, after recalling the basic facts regarding Parrondo games and the FNA quantization protocols, proper quantizations for the history dependent Parrondo game and their randomized sequences are constructed.

3.1 Parrondo Games

Parrondo et. al first formulated such games in [29]. The subject of Parrondo games has seen much research activity since then. Parrondo games typically involve the flipping of biased coins and yield only expected payoffs. A Parrondo game whose expected payoff is positive is said to be *winning*. If the expected payoff is negative, the game is said to be *losing*, and if the expected payoff is 0, the game is said to be *fair*.

Parrondo games are of interest because sequences of such games occasionally exhibit the *Parrondo effect*; that is, when two or more losing games are appropriately sequenced, the resulting combined game is winning. Frequently, this sequence is *randomized* which means that the game played at each stage of the sequence is chosen at random with respect to a particular probability distribution over the games being sequenced. A comprehensive survey of Parrondo games and the Parrondo effect by Abbott and Harmer can be found in [14].

Earlier work on the quantization of Parrondo games can be found in [21] where Meyer offers an analysis of a quantization of a particular type of Parrondo game, and in [11] where Abbott, Flitney, and Ng (AFN) propose quantizations of a different type of Parrondo game. The authors of both papers quantize their original game via their own particular quantization protocols, and further, model the game sequences as iterations of their protocols. In each of these protocols, quantum actions are performed on a collection of initial states of a quantum system. At the end, a measurement of certain specific states is made and, from the resulting probability distributions, an expected payoff computed.

3.1.1 Capital Dependent Parrondo Games

In [29], Parrondo et al describe two types of coin flipping games which have the property that if individually repeated, the games result in a decreasing expected payoff to the player, yet when the two games are played in a deterministic or probabilistic sequence repeatedly, the expected payoff to the player increases over time.

Suppose that $X(t) = 0, 1, 2, \ldots$ is the capital available to the player. If the player wins a game, then the capital increases by one, and if the player loses, then the capital

Table 3.1: Game B

	Prob. of gain	Prob. of loss
$X(t) \equiv 0 \bmod 3$	p_1	$1 - p_1$
$X(t) \equiv 1$ or $2 \bmod 3$	p_2	$1 - p_2$

decreases by one. The simplest type of this game, referred to in the literature as game A, is determined by a biased coin with probability of gain p. That is, the capital increases by one with probability p and decreases by one with probability $1 - p$. Another game, called game B, is defined by two biased coins. The choice of which coin is to be played in an instance of the game B is determined by the congruence modulo 3 of the capital, $X(t)$, available to the player in that instance. Hence, game B is defined by the rules given in table 3.1.

Parrondo et al set $p = \frac{1}{2} - \epsilon$, $p_1 = \frac{1}{10} - \epsilon$, $p_2 = \frac{3}{4} - \epsilon$, for $\epsilon > 0$ as an example of games A and B which are losing if played individually or in a fixed sequence, but which, when combined in a randomized sequence with the uniform distribution over the two games, is winning. Both games A and B are losing, winning, or fair as $\epsilon > 0$, $\epsilon < 0$ and $\epsilon = 0$, respectively. Parrondo et al consider in detail the case when both games A and B are fair. The game B is analyzed as a Markov process $Y(t) \equiv X(t)$ (mod 3), that is, $Y(t)$ is equal to the remainder upon dividing the capital $X(t)$ by 3. A transition matrix for game B is thus given by

$$T = \begin{pmatrix} 0 & \frac{1}{4} & \frac{3}{4} \\ \\ \frac{1}{10} & 0 & \frac{1}{4} \\ \\ \frac{9}{10} & \frac{3}{4} & 0 \end{pmatrix}. \tag{3.1}$$

The stationary state for this Markov process can be computed from the matrix equation

$$\begin{pmatrix} 0 & \frac{1}{4} & \frac{3}{4} \\ \\ \frac{1}{10} & 0 & \frac{1}{4} \\ \\ \frac{9}{10} & \frac{3}{4} & 0 \end{pmatrix} \begin{pmatrix} \pi_0 \\ \\ \pi_1 \\ \\ \pi_2 \end{pmatrix} = \begin{pmatrix} \pi_0 \\ \\ \pi_1 \\ \\ \pi_2 \end{pmatrix} \tag{3.2}$$

where π_i is the probability of the capital $X(t)$ taking on a value congruent to i (mod 3), $i = 0, 1, 2$. The matrix Equation (3.2) gives rise to the following system of equations

$$\frac{1}{4}\pi_1 + \frac{3}{4}\pi_2 = \pi_0 \tag{3.3}$$

$$\frac{1}{10}\pi_0 + \frac{1}{4}\pi_2 = \pi_1 \tag{3.4}$$

$$\frac{9}{10}\pi_0 + \frac{3}{4}\pi_1 = \pi_2 \tag{3.5}$$

which has the following solution.

$$\pi_0 = \pi_0, \quad \pi_1 = \frac{2}{5}\pi_0, \quad \pi_2 = \frac{6}{5}\pi_0.$$

Since the game is assumed to be fair, $p_1\pi_0 + p_2\pi_1 + p_2\pi_2 = \frac{1}{2}$, and one computes $\pi_0 = \frac{5}{13}$, $\pi_1 = \frac{2}{13}$, $\pi_2 = \frac{6}{13}$.

Now if the fair games A and B are played in a randomized sequence, the resulting capital can be increasing. To see this, let q be the probability with which the game A is played. Then game B is played with probability $(1-q)$. Again, analyze the Markov sequence $Y(t) \equiv X(t)$ (mod 3), but this time the transition matrix is

$$T' = \begin{pmatrix} 0 & \frac{1}{2}q + \frac{1}{4}(1-q) & \frac{1}{2}q + \frac{3}{4}(1-q) \\ \frac{1}{2}q + \frac{1}{10}(1-q) & 0 & \frac{1}{2}q + \frac{1}{4}(1-q) \\ \frac{1}{2}q + \frac{9}{10}(1-q) & \frac{1}{2}q + \frac{3}{4}(1-q) & 0 \end{pmatrix} \tag{3.6}$$

To sequence these games via the uniform distribution, set $q = \frac{1}{2}$ and get

$$T' = \begin{pmatrix} 0 & \frac{1}{4} + \frac{1}{8} & \frac{1}{4} + \frac{3}{8} \\ \frac{1}{4} + \frac{1}{20} & 0 & \frac{1}{4} + \frac{1}{8} \\ \frac{1}{4} + \frac{9}{20} & \frac{1}{4} + \frac{3}{8} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{3}{8} & \frac{5}{8} \\ \frac{3}{10} & 0 & \frac{3}{8} \\ \frac{7}{10} & \frac{5}{8} & 0 \end{pmatrix} \tag{3.7}$$

Before last $t-2$	Last $t-1$	Coin	Prob. of gain at t	Prob. of loss at t
gain	gain	B_1'	p_1	$1-p_1$
gain	loss	B_2'	p_2	$1-p_2$
loss	gain	B_3'	p_3	$1-p_3$
loss	loss	B_4'	p_4	$1-p_4$

Table 3.2: History dependent game B'.

Computing the stationary state $(\pi_0', \pi_1', \pi_2')^T$ for the case in which each game A and B is fair, gives $\pi_0' = \frac{245}{709}$, $\pi_1' = \frac{180}{709}$, and $\pi_2' = \frac{284}{709}$ up to a normalization constant. Note that $\pi_0' = \frac{245}{709}$ is larger than $\frac{5}{13}$, and thus the capital increases.

3.1.2 A History Dependent Parrondo Game

The history dependent Parrondo game, introduced in [29] by Parrondo et al, is again a biased coin flipping game, where now the choice of the biased coin depends on the history of the game thus far, as opposed to the modular value of the capital. A history dependent Parrondo game B' with a two stage history is reproduced in Table 3.2.

As above, let $X(t)$ be the capital available to the player at time t. At stage t, this capital goes up or down by one unit, the probability of gain determined by the biased coin used at that stage. Obtain a Markov process by setting

$$Y(t) = \begin{pmatrix} X(t) - X(t-1) \\ X(t-1) - X(t-2) \end{pmatrix}. \tag{3.8}$$

This allows one to analyze the long term behavior of the capital in game B' via the stationary state of the process $Y(t)$. The transition matrix for this process is

$$X = \begin{pmatrix} p_1 & 0 & p_3 & 0 \\ 1-p_1 & 0 & 1-p_3 & 0 \\ 0 & p_2 & 0 & p_4 \\ 0 & 1-p_2 & 0 & 1-p_4 \end{pmatrix} \tag{3.9}$$

The stationary state can be computed from the following equations

$$p_1\pi_1 + p_3\pi_3 = \pi_1$$

$$(1-p_1)\pi_1 + (1-p_3)\pi_3 = \pi_2$$

$$p_2\pi_2 + p_4\pi_4 = \pi_3$$

$$(1-p_2)\pi_2 + (1-p_4)\pi_4 = \pi_4$$

and is given by

$$s = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix} = \frac{1}{N}\begin{pmatrix} p_3p_4 \\ p_4(1-p_1) \\ p_4(1-p_1) \\ (1-p_1)(1-p_2) \end{pmatrix} \tag{3.10}$$

after setting the free variable $v_4 = (1-p_1)(1-p_2)$ and normalization constant

$$N = \sqrt{\sum_{j=1}^{4}(\pi_j)^2} = \sqrt{(p_3p_4)^2 + 2\left[(1-p_1)p_4\right]^2 + \left[(1-p_1)(1-p_2)\right]^2}$$

which simplifies to

$$N = (1 - p_1)(2p_4 + 1 - p_2) + p_3 p_4.$$

Consequently, the probability of gain in a generic run of the game B' is

$$p_{\text{gain}}^{B'} = \frac{1}{N} \sum_{j=1}^{4} \pi_j p_j = \frac{p_4 \left(p_3 + 1 - p_1\right)}{\left(1 - p_1\right)\left(2p_4 + 1 - p_2\right) + p_3 p_4} \tag{3.11}$$

where π_j is the probability that a certain history j, represented in binary format, will occur, while p_j is the probability of gain upon the flip of the last coin corresponding to history j. The expression for $p_{\text{gain}}^{B'}$ simplifies to

$$p_{\text{gain}}^{B'} = 1/(2 + x/y) \tag{3.12}$$

with

$$y = p_4(p_3 + 1 - p_1) > 0 \tag{3.13}$$

for any choice of the probabilities $p_1, \ldots p_4$, and

$$x = (1 - p_1)(1 - p_2) - p_3 p_4. \tag{3.14}$$

Therefore, game B' obeys the following rule: if $x < 0$, B' is winning, that is, has positive expected payoff; if $x = 0$, B' is fair; and if $x > 0$, B' is losing, that is, has negative expected payoff.

Before proceeding further, it is useful to view the preceding ideas in a more formal game theoretic context. For this, consider the Parrondo games as one player games in normal form, that is, as a function, where the one player's strategic choices in part correspond to the biases of the coins. For a history dependent Parrondo game with two historical stages, Parrondo et al refer to these choices as a "choice of rules." However, the mere choice of biases for the coins is not enough to determine a unique normal form for these history dependent Parrondo games. In particular, an initial probability distribution over the allowable histories is also required. Although any specific distribution suffices to uniquely determine such a normal form, as the structure of the game is given by a Markov process, there is a natural choice for this initial distribution. Though this issue is not discussed by Parrondo et al, these authors immediately focus on this natural choice, namely, the distribution corresponding to the stationary state of the Markov process representing the game.

Now, the normal form of these history dependent Parrondo games maps the tuple (P, s) into the element

$$(\pi_1 p_1, \pi_1(1-p_1), \pi_2 p_2, \pi_2(1-p_2), \pi_3 p_3, \pi_3(1-p_3), \pi_4 p_4, \pi_4(1-p_4))$$

of the probability payoff space $[0,1]^{\times 8}$, where $s = (\pi_1, \pi_2, \pi_3, \pi_4) \in \Delta(\text{hist}G)$ is the stationary state of the Markov process with transition matrix defined by $P = (p_1, p_2, p_3, p_4)$ $\in [0,1]^{\times 4}$, as in Equation (3.9). Formally,

$$G_s : [0,1]^{\times 4} \times \Delta(\text{hist}G) \rightarrow [0,1]^{\times 8} \tag{3.15}$$

$$G_s : (P, s) \mapsto (\pi_1 p_1, \pi_1(1-p_1), \pi_2 p_2, \pi_2(1-p_2), \pi_3 p_3, \pi_3(1-p_3), \pi_4 p_4, \pi_4(1-p_4)) \tag{3.16}$$

The outcomes *winning*, *breaking even*, or *losing* to the player occur when $p^{B'}_{\text{gain}} > \frac{1}{2}$, $p^{B'}_{\text{gain}} = \frac{1}{2}$, and $p^{B'}_{\text{gain}} < \frac{1}{2}$, respectively.

Note that in this more formal game theoretic context for history dependent Parrondo games, the dependence of these games on the initial probability distribution s is made clear. This initial probability distribution plays the role of the initial state s for the classical game G_s appearing in the proper quantization discussion at the end of chapter 2.

3.1.3 Randomized Combinations of History Dependent Parrondo Games

Consider now the two stage history dependent game obtained by randomly sequencing the games B' and B'' where each of B' and B'' are history dependent Parrondo games with two stage histories. This can be formally considered as a real convex linear combination of the games B' and B'', where the coefficients on B' and B'' are given by r, the probability that the game B' is played at a given stage, and $(1-r)$, the probability that the game B'' is played at a given stage. This is because the transition matrix of the Markov process associated to the randomized sequence is obtained from the transition matrices T' and T'' for the games B' and B'', respectively, by taking the real convex combination $rT' + (1-r)T''$. Explicitly, let

$$T' = \begin{pmatrix} \alpha_1 & 0 & \alpha_3 & 0 \\ 1-\alpha_1 & 0 & 1-\alpha_3 & 0 \\ 0 & \alpha_2 & 0 & \alpha_4 \\ 0 & 1-\alpha_2 & 0 & 1-\alpha_4 \end{pmatrix} \tag{3.17}$$

and

$$T'' = \begin{pmatrix} \beta_1 & 0 & \beta_3 & 0 \\ 1-\beta_1 & 0 & 1-\beta_3 & 0 \\ 0 & \beta_2 & 0 & \beta_4 \\ 0 & 1-\beta_2 & 0 & 1-\beta_4 \end{pmatrix}. \tag{3.18}$$

with $\alpha_j, \beta_j \in [0,1]$ representing the probability of gain for the j coin in games B' and B'' respectively. Then the transition matrix $rT' + (1-r)T''$ of the Markov process for the randomized sequence of B' and B'' consists of entries $t_j = r\alpha_j + (1-r)(\beta_j)$ and $1 - t_j = r(1-\alpha_j) + (1-r)(1-\beta_j)$ in the appropriate locations. Call this randomized sequence of games B' and B'' the history dependent game $B'B''$ with probability of gain t_j. The stable state, computed in exactly the same fashion as the stable state for the game B' in section 3.1.2 above, has form

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} = \frac{1}{R}\begin{pmatrix} t_3t_4 \\ t_4(1-t_1) \\ t_4(1-t_1) \\ (1-t_1)(1-t_2) \end{pmatrix} \tag{3.19}$$

with $R = \sum_{j=1}^{4} \tau_j$ a normalization constant. Using the stable state, the probability of gain in the game $B'B''$ is computed to be

$$p_{\text{gain}}^{B'B''} = \frac{1}{R}\sum_{j=1}^{4} \tau_j t_j = \frac{t_4\,(t_3 + 1 - t_1)}{(1-t_1)\,(2t_4 + 1 - t_2) + t_3t_4}. \tag{3.20}$$

Just as in case of the game B', the expression for $p_{\text{gain}}^{B'B''}$ reduces to

$$p_{\text{gain}}^{B'B''} = 1/(2 + x'/y') \tag{3.21}$$

with

$$y' = t_4(t_3 + 1 - t_1) > 0 \tag{3.22}$$

for any choice of the probabilities $t_1, \ldots t_4$, and

$$x' = (1 - t_1)(1 - t_2) - t_3 t_4. \tag{3.23}$$

The game $B'B''$ therefore behaves entirely like the game B', following the rule: if $x' < 0$, $B'B''$ is winning, that is, has positive expected payoff; if $x' = 0$, $B'B''$ is fair; and $x' > 0$, $B'B''$ is losing, that is, has negative expected payoff.

It is therefore possible to adjust the values of the α_j and β_j in games B' and B'' so that they are individually losing, but the combined game $B'B''$ is now winning. This is the Parrondo effect. In the present example, the Parrondo effect occurs when

$$(1 - \alpha_3)(1 - \alpha_4) > \alpha_1 \alpha_2 \tag{3.24}$$

$$(1 - \beta_3)(1 - \beta_4) > \beta_1 \beta_2 \tag{3.25}$$

and

$$(1 - t_3)(1 - t_4) < t_1 t_2. \tag{3.26}$$

The reader is referred to [15] for a detailed analysis of the values of the parameters which lead to the Parrondo effect in such games.

Restricting to the original work of Parrondo et al, a special case occurs when we consider one of the games in the randomized sequence to be of type A. That is, flipping a single biased coin which on the surface appears to have no history dependence. However, note that such a game may be interpreted as a history dependent Parrondo game with a two stage history where the coin used in A is employed for every history. Call such a history dependent game A'. The transition matrix for A' takes the form

$$\Delta = \begin{pmatrix} p & 0 & p & 0 \\ 1-p & 0 & 1-p & 0 \\ 0 & p & 0 & p \\ 0 & 1-p & 0 & 1-p \end{pmatrix}. \tag{3.27}$$

Now, forming randomized sequences of games A' and B' is seen to agree with the forming of convex linear combinations mentioned above. In particular, as analyzed in [30] if games A' and B' are now sequenced randomly with equal probability, the Markov process for the randomized sequence is given with transition matrix containing the entries $q_j = \frac{1}{2}(\alpha_j + p)$ and $1 - q_j = \frac{1}{2}[(1-\alpha_j) + (1-p)]$ in the appropriate locations (recall that the probability of win for game A is p), and has stationary state

$$\rho = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{pmatrix} = \frac{1}{M} \begin{pmatrix} q_3 q_4 \\ q_4(1-q_1) \\ q_4(1-q_1) \\ (1-q_1)(1-q_2) \end{pmatrix} \tag{3.28}$$

Denote this randomized sequence of games A' and B' by $A'B'$. The probability of gain in the game $A'B'$ is

$$p_{\text{gain}}^{A'B'} = \frac{1}{M}\sum_{j=1}^{4} \rho_j q_j = \frac{q_4\,(q_3+1-q_1)}{(1-q_1)\,(2q_4+1-q_2)+q_3q_4} \tag{3.29}$$

As in the more general case of the game $B'B''$, it is now possible to adjust the values of the parameters p and p_j's in games A' and B' so that they are individually losing, but the combined game $A'B'$ is now winning. This happens when

$$1-p>p \tag{3.30}$$

$$(1-\alpha_3)(1-\alpha_4) > \alpha_1\alpha_2 \tag{3.31}$$

and

$$(1-q_3)(1-q_4) < q_1q_2. \tag{3.32}$$

Parrondo et al show in [30] that when $p = \frac{1}{2} - \epsilon$, $\alpha_1 = \frac{9}{10} - \epsilon$, $\alpha_2 = \alpha_3 = \frac{1}{4} - \epsilon$, $\alpha_4 = \frac{7}{10} - \epsilon$, and $\epsilon < \frac{1}{168}$, the inequalities (3.30)-(3.32) are satisfied. This is Parrondo et al's original example of the Parrondo effect for history dependent Parrondo games.

3.2 The FNA Quantization of Parrondo Games

In [11], Flitney, Ng, and Abbott quantize the type A' Parrondo game by considering the action of an element of $SU(2)$ on a qubit and interpret this as "flipping" a biased quantum coin. They consider history dependent games with $(n-1)$ stage histories, and in the language of the Bleiler formalism, quantize these games via a family of protocols. In every protocol, n qubits are required and the unitary operator representing the entire game is a $2^n \times 2^n$ block diagonal matrix with the 2×2 blocks composed of arbitrary elements of $SU(2)$. In the language of quantum logic circuits, this is a quantum multiplexer [17]. The first $(n-1)$ qubits represent the history of the game via controls, as illustrated in Figure 3.1 for a two stage history dependent game similar to the game B' given in Table 3.2. Each protocol is defined as the action of the quantum multiplexer on the n qubits.

The quantum multiplexer illustrated in Figure 3.1, where the elements $Q_1 \ldots Q_4$ are elements of $SU(2)$, operates as follows. When the basis of the state space $(\mathbb{C}P^1)^{\otimes 3}$ of three qubits is the computational basis

$$\mathcal{B} = \{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}.$$

the quantum multiplexer takes on the form of an 8×8 block diagonal matrix of the form

$$Q = \begin{pmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ 0 & 0 & Q_3 & 0 \\ 0 & 0 & 0 & Q_4 \end{pmatrix}, \tag{3.33}$$

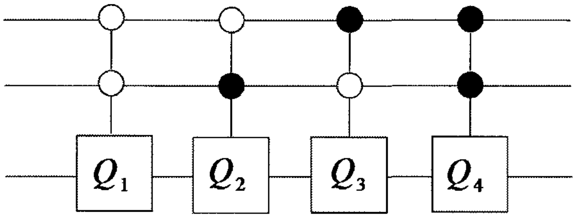


Figure 3.1: Part of the quantization protocol for the history dependent Parrondo game. The first two wires represent the history qubits.

where each $Q_j \in SU(2)$. That is

$$Q_j = \begin{pmatrix} a_j & -\bar{b}_j \\ b_j & \bar{a}_j \end{pmatrix} \tag{3.34}$$

with $a_j, b_j \in \mathbb{C}$ satisfying $|a_j|^2 + |b_j|^2 = 1$.

For further description of the workings of the quantum multiplexer, the following convention, found in D. Meyer's original work [20], will be used. Let a "win" or "gain" for a player be represented by the action "No Flip" which is the identity element of $SU(2)$. For example, in Meyer's quantum penny flip game, the "quantum coin" is in the initial state of "Head" represented by $|0\rangle$ and a gain for the player using the quantum strategies occurs when the final orientation state of the coin is observed to be $|0\rangle$. This is contrast to the convention in FNA [11] where $|1\rangle$ represents a gain.

Now the first two qubits of an element of $\mathcal{B}$ represent a history of the classical game, with $|0\rangle$ representing gain (G) and the $|1\rangle$ representing loss (L). The blocks Q_j act on the third qubit in the circuit under the control of the history represented by the binary configuration of the first two qubits. For example, if the first two qubits are in the joint state $|00\rangle$, the $SU(2)$ action Q_1 is applied to the third qubit. Similarly, for the other three basic initial joint states of the first two qubits. This models the historical dependence of the game by having the history (G, G) correspond to the initial joint state $|00\rangle$ of the first two qubits, the history (G, L) correspond to the initial joint state $|01\rangle$, the history (L, G) correspond to the initial joint state $|10\rangle$, and the history (L, L) correspond to the initial joint state $|11\rangle$. Thus, an appropriate action is taken for each history.

Recall from section 3.1.2 that the evaluation of the behavior of the classical history dependent Parrondo game requires more than just the Markov process. The evaluation also requires the stable state and a payoff rule. Note that the results of applying the quantum multiplexer depends entirely on the initial state on which it acts. That is, different initial states result in differing final states. The payoff rule used by Abbott, Flitney, and Ng resembles that for the classical game in that the quantized versions are winning when the expectation greater than 0 (gain capital), fair if the expectation is equal to 0 (break even), and losing if the expectation is less than 0 (lose capital). Further, as in the classical game this question is decided by examining the probability of gain versus the probability of loss. In particular, if the probability of gain is greater than $\frac{1}{2}$, the quantum game is winning.

3.2.1 Problems with the FNA Protocol

The FNA quantization protocols for the history dependent game attempt to replace the classical biases of the coins in the game with arbitrary elements of $SU(2)$ and the stable state of Markov process describing the dynamics of the game with certain initial states of the qubits on which a quantum multiplexer, composed of the arbitrary elements of $SU(2)$, acts. The problems with the FNA quantization protocols are two-fold. First, the attempted embedding of the classical history dependent game into the quantized game by replacing the biases of the classical coins with arbitrary $SU(2)$ elements, turns out to be *relational* rather than functional. That is, Equations (3.33) and (3.34) together give a family of quantum multiplexers that the classical game maps into via the embedding. This relational mapping makes it impossible to recover the classical game by restricting the quantized game to the image of the embedding. Therefore, the FNA quantization of the history dependent Parrondo game is not proper.

The second problem arises from the choice of initial state. No attempt is made to produce an analog of the stable state of a Markov process. Instead, the authors mention the obvious fact that different initial states will produce different results, and in particular consider two arbitrary initial states, one the maximally entangled state $\frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right)$, the other the basic state $|000\rangle$. In the latter, the authors assert that the quantum game behaves like a classical game with fixed initial history (L, L), according to their convention in which $|0\rangle$ represents loss. Note that even if the this is not a proper quantization of any classical history dependent game as it fails to incorporate the other histories represented in the stable state. For

$$|000\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and when acted upon by the quantum multiplexer in Equation (3.33) produces the output

$$\begin{pmatrix} a_1 \\ b_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

which makes the failure of the protocol to incorporate the other histories apparent.

In the former, a similar situation occurs where only the histories $|000\rangle$ and $|111\rangle$ are incorporated. This protocol is also not proper as only the histories (L, L) and (G, G) are non-trivially represented in the initial state. For

$$\frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

and when acted upon by the quantum multiplexer in Equation (3.33) produces the output

$$\frac{1}{\sqrt{2}}\begin{pmatrix} a_1 \\ b_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\overline{b_4} \\ \overline{a_4} \end{pmatrix}$$

from which, again, the failure of the protocol to incorporate the other histories is apparent.

Moreover, both quantization protocols fail to reproduce the Markovian dynamics and the payoff function of the original game.

Flitney et al also consider various “sequences” of the quantum games A' and B', where B' is played with three qubits and quantized using the maximally entangled initial state. These sequences are defined by compositions of the unitary operators defining the games. Indeed, these sequences now produce the results presented in [11]. These results are certainly novel and perhaps carry scientific significance; however, they fail to carry game-theoretic significance as, with respect to the classical Parrondo games, each arises from a quantization that is *not* proper.

In light of the Bleiler formalism discussed in chapter 2, constructing proper quantizations of games is a fundamental problem for quantum theory of games. In the following section, a proper quantization paradigm is developed for both history dependent Parrondo games and randomized sequences of such.

3.3 Properly Quantizing History Dependent Parrondo Games

Consider the history dependent game B' with only 2 histories. As in the FNA protocol, the quantization protocol for this game uses a three qubit quantum multiplexer with matrix representation

$$Q = \begin{pmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ 0 & 0 & Q_3 & 0 \\ 0 & 0 & 0 & Q_4 \end{pmatrix}$$

with each $Q_j \in SU(2)$, together with an initial state.

To reproduce the classical game, first embed the four classical coins that define the game B' into blocks of the matrix Q corresponding to the appropriate history. The embedding is via superpositions of the embeddings of the classical actions of "No Flip" and "Flip" on the coins into $SU(2)$ given either by

$$N = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad F = \begin{pmatrix} 0 & -\bar{\eta} \\ \eta & 0 \end{pmatrix} \tag{3.35}$$

or by

$$N^{*} = \begin{pmatrix} i & 0 \\ 0 & \bar{i} \end{pmatrix}, \quad F^{*} = \begin{pmatrix} 0 & -\overline{i\eta} \\ i\eta & 0 \end{pmatrix} \tag{3.36}$$

with $\eta^6 = 1$. Call the embeddings in equations (3.35) *basic embeddings of type 1* and the embedding in equations (3.36) *basis embeddings of type 2*. Choosing the basic embedding of type 1 embeds the j^{th} coin into $SU(2)$ as

$$Q_j = \sqrt{p_j}N + \sqrt{(1-p_j)}F = \begin{pmatrix} \sqrt{p_j} & -\sqrt{1-p_j}\bar{\eta} \\ \sqrt{1-p_j}\eta & \sqrt{p_j} \end{pmatrix} \tag{3.37}$$

where p_j is the probability of gain when the j^{th} coin is played in the classical game B' given in Table 3.2. Note that the probabilities p_j of gaining are associated with the classical action N in line with Meyer's original convention from [20] where $|0\rangle$ represents a gain. Hence, the elements of the subset

$$\mathcal{W} = (|000\rangle, |010\rangle, |100\rangle, |110\rangle)$$

of $\mathcal{B}$ all represent possible gaining outcomes in the game. The probability of gain in the quantized game is therefore the sum of the coefficients of the elements of $\mathcal{W}$ that result from measurement.

Next, set the initial state I equal to

$$\frac{1}{\sqrt{\sum_{j=1}^{n} \pi_j}} \begin{pmatrix} \sqrt{\pi}_1 \\ 0 \\ \sqrt{\pi}_2 \\ 0 \\ \sqrt{\pi}_3 \\ 0 \\ \sqrt{\pi}_4 \\ 0 \end{pmatrix}, \tag{3.38}$$

where the π_j are the probabilities with which the histories occur in the classical game, as computed from the stationary state of the Markovian process of section 3.1.2. The quantum multiplexer Q acts on I to produce the final state

$$F_I = \frac{1}{\sqrt{\sum_{j=1}^{4} \pi_j}} \begin{pmatrix} \sqrt{p_1\pi_1} \\ \eta\sqrt{(1-p_1)\pi_1} \\ \sqrt{p_2\pi_2} \\ \eta\sqrt{(1-p_2)\pi_2} \\ \sqrt{p_3\pi_3} \\ \eta\sqrt{(1-p_3)\pi_3} \\ \sqrt{p_4\pi_4} \\ \eta\sqrt{(1-p_4)\pi_4} \end{pmatrix}. \tag{3.39}$$

Measuring the state F_I in the observational basis and adding together the resulting coefficients of the elements of the set $\mathcal{W}'$ gives the probability of gain in the quantized game to be

$$p_{\text{gain}}^{QB'} = \frac{1}{\sum_{j=1}^{4} \pi_j} \left(\sum_{j=1}^{4} p_j \pi_j \right) = \frac{1}{N} \left(\sum_{j=1}^{4} p_j \pi_j \right) \tag{3.40}$$

which is equal to the probability of gain in the classical game.

This proper quantization paradigm is based on the philosophy first discussed at the end of chapter 2. That is, a proper quantization of a classical game G_s that depends on an initial state s requires that s be embedded into an initial state I on which the quantum multiplexer acts. Here, the initial state $s = (\pi_1, \pi_2, \pi_3, \pi_4) \in [0,1]^{\times 4}$ embeds as the initial state $I \in (\mathbb{C}P^1)^{\otimes 3}$ given in expression (3.38). The resulting game $G_s^{\Theta_I}$ is the quantization of the classical game G_s by the protocol Θ_I which maps the tuple (Q, I), with $Q = (Q_1, Q_2, Q_3, Q_4) \in [SU(2)]^{\times 4}$ to $F_I \in (\mathbb{C}P^1)^{\otimes 3}$ given in Equation (3.39).

$$\begin{tikzcd}
(Q,I) \arrow[r, "\Theta_I"] \arrow[rrd, "G_s^{\Theta_I}"] & \left(CP^1\right)^{\otimes 3} \arrow[r, "\mathrm{Proj}"] & QS(\mathrm{Im}\,G) \arrow[d, "q_{meas}"] \\
(S,s) \arrow[u, dashed, "e"] \arrow[rr, "G_s"] & & \Omega
\end{tikzcd}$$

Figure 3.2: Proper Quantization, using the embedding e, of the History Dependent Game via the quantization protocol Θ_I.

Formally,

$$\Theta_I : [SU(2)]^{\times 4} \times (\mathbb{C}P^1)^{\otimes 3} \to (\mathbb{C}P^1)^{\otimes 3} \tag{3.41}$$

$$\Theta_I : (Q, I) \mapsto F_I \tag{3.42}$$

By projecting on to the gaining basis $\mathcal{W}$, one now gets a quantum superposition over the image ImG of the game G. Finally, quantum measurement produces ImG. Call $Proj$ the function that projects F_I on to $\mathcal{W}$, and denote quantum measurement by q_{meas}. Then

$$G_s^{\Theta_I} = q_{meas} \circ Proj \circ \Theta_I : (Q, I) \mapsto \mathrm{Im}G \tag{3.43}$$

is a proper quantization of the payoff function of the normal form of classical history dependent game G_s given in Equations (3.15) and (3.16). Equation (3.43) can be expressed by the commutative diagram of Figure 3.2, which the reader is urged to compare and contrast with Figure 2.4 in chapter 2.

Note that by embedding s into I, the notion of randomization via probability distributions is generalized in the quantum game to the higher order notion of randomization via quantum superpositions plus measurement. In particular, the probability distribution $P = (p_1, p_2, p_3, p_4) \in [0,1]^{\times 4}$ that defines the Markov process associated with the game is replaced with the quantum multiplexer $Q = (Q_1, Q_2, Q_3, Q_4) \in [SU(2)]^{\times 4}$ associated with the quantized game, and the stable state s of the Markov process is replaced with an initial evaluative state I of the quantum multiplexer.

3.4 Properly Quantizing Randomized Sequences of History Dependent Parrondo Games

Recall from section 3.1.3 that randomized sequences of games B' and B'' are analyzed via a Markov process with transition matrix equal to a real convex combination of the transition matrices of each game in which B' is played with probability r and B'' with probability $(1-r)$. Moreover, such a sequence is considered to by an instance of a history dependent game denoted as $B'B''$.

Motivated by the discussion on proper quantization of the game Parrondo games B' and B'' in section 3.3 above, let us now consider a higher order randomization in the form of a quantum superposition of the quantum multiplexers used in the proper quantization of the the games B' and B'' with the goal of producing a proper quantization of the game $B'B''$.

As in section 3.3, associate the quantum multiplexer $Q' = (Q'_1, Q'_2, Q'_3, Q'_4)$ with the game B', where

$$Q'_j = \sqrt{\alpha_j} N + \sqrt{(1-\alpha_j)} F = \begin{pmatrix} \sqrt{\alpha_j} & -\sqrt{1-\alpha_j}\,\bar{\eta} \\ \sqrt{1-\alpha_j}\,\eta & \sqrt{\alpha_j} \end{pmatrix},$$

Next, associate the quantum multiplexer $Q'' = (Q''_1, Q''_2, Q''_3, Q''_4)$ with the game B'', where

$$Q''_j = \sqrt{\beta_j}N^* + \sqrt{(1-\beta_j)}F^* = \left(\begin{array}{cc} \sqrt{\beta}_j i & -\sqrt{1-\beta_j}(\overline{i\eta}) \\ \sqrt{1-\beta_j} i\eta & \sqrt{\beta}_j \bar{i} \end{array}\right).$$

Now consider the quantum superposition

$$\Sigma = \gamma' Q' + \gamma'' Q'' \tag{3.44}$$

$$= \left(\begin{array}{llll} \gamma' Q'_1 + \gamma'' Q''_1 & 0 & 0 & 0 \\ 0 & \gamma' Q'_2 + \gamma'' Q''_2 & 0 & 0 \\ 0 & 0 & \gamma' Q'_3 + \gamma'' Q''_3 & 0 \\ 0 & 0 & 0 & \gamma' Q'_4 + \gamma'' Q''_4 \end{array}\right) \tag{3.45}$$

of the quantum multiplexers Q' and Q'' with

$$(\gamma')^2 + (\gamma'')^2 = 1, \quad |\gamma'|^2 = r, \quad |\gamma''|^2 = (1-r), \quad \overline{\gamma'}\gamma'' - \overline{\gamma''}\gamma' = 0 \tag{3.46}$$

and

$$\gamma' Q'_j + \gamma'' Q''_j = \left(\begin{array}{cc} \gamma'\sqrt{\alpha}_j + \gamma''\sqrt{\beta_j} i & -\left(\gamma'\sqrt{1-\alpha_j} - \gamma''\sqrt{1-\beta_j} i\right)\overline{\eta} \\ \left(\gamma'\sqrt{1-\alpha_j} + \gamma''\sqrt{1-\beta_j} i\right)\eta & \gamma'\sqrt{\alpha}_j - \gamma''\sqrt{\beta_j} i \end{array}\right) \tag{3.47}$$

Set the evaluative initial state in this case equal to

$$I = \frac{1}{\sqrt{\sum_{j=1}^{n} \tau_j}} \begin{pmatrix} \sqrt{\tau}_1 \\ 0 \\ \sqrt{\tau}_2 \\ 0 \\ \sqrt{\tau}_3 \\ 0 \\ \sqrt{\tau}_4 \\ 0 \end{pmatrix} \tag{3.48}$$

where the τ_j are the probabilities that form the stationary state of the classical game $B'B''$ given in Equation (3.19). The claim is that the quantum multiplexer Σ in Equation (3.44) together with the evaluative initial state I in Equation (3.52) define a proper quantization of the classical game $B'B''$ in which B' is played with probability r and and B'' is played with probability $(1 - r)$.

To check the validity of this claim, compute the output of Σ for the evaluative initial state I in Equation (3.52):

$$\frac{1}{\sqrt{\sum_{j=1}^{n}\tau_j}}\begin{pmatrix} \sqrt{\tau}_1(\gamma'\sqrt{\alpha}_1+\gamma''\sqrt{\beta_1}i) \\ \sqrt{\tau}_1\left(\gamma'\sqrt{1-\alpha_1}+\gamma''\sqrt{1-\beta_1}i\right)\eta \\ \sqrt{\tau}_2(\gamma'\sqrt{\alpha}_2+\gamma''\sqrt{\beta_2}i) \\ \sqrt{\tau}_2\left(\gamma'\sqrt{1-\alpha_2}+\gamma''\sqrt{1-\beta_2}i\right)\eta \\ \sqrt{\tau}_3(\gamma'\sqrt{\alpha}_3+\gamma''\sqrt{\beta_3}i) \\ \sqrt{\tau}_3\left(\gamma'\sqrt{1-\alpha_3}+\gamma''\sqrt{1-\beta_3}i\right)\eta \\ \sqrt{\tau}_4(\gamma'\sqrt{\alpha}_4+\gamma''\sqrt{\beta_4}i) \\ \sqrt{\tau}_4\left(\gamma'\sqrt{1-\alpha_4}+\gamma''\sqrt{1-\beta_4}i\right)\eta \end{pmatrix}.$$

The probability of gain produced upon measurement of this output is

$$p_{\text{gain}}^{QB'B''}=\frac{1}{\sum_{j=1}^{n}\tau_j}\sum_{j=1}^{4}\left|\sqrt{\tau}_j(\gamma'\sqrt{\alpha}_j+\gamma''\sqrt{\beta_j}i)\right|^2 \tag{3.49}$$

which simplifies to

$$\frac{1}{R}\sum_{j=1}^{4}\tau_j\left[|\gamma'|^2\alpha_j+|\gamma''|^2\beta_j+\sqrt{\alpha_j\beta_j}i\left(\overline{\gamma'}\gamma''-\overline{\gamma''}\gamma'\right)\right]. \tag{3.50}$$

Using the conditions set up in Equation (3.46), the previous expression further simplifies to give

$$p_{\text{gain}}^{QB'B''}=\frac{1}{R}\sum_{j=1}^{4}\tau_j\left[r\alpha_j+(1-r)\beta_j\right]=\frac{1}{R}\sum_{j=1}^{4}\tau_jt_j.$$

which is exactly that given in Equation (3.20) in section 3.4 for the classical game $B'B''$.

Again, note that this proper quantization paradigm requires mapping of the initial state of the classical game $B'B''$, which is a probability distribution, into an initial state which the quantization protocol acts on, which is a higher order randomization in the form of a quantum superposition which measures appropriately with respect to the observational basis. The image of the normal form of the quantum game in $[0, 1]$ agrees precisely with $p_{\text{gain}}^{QB'B''}$. Note that in this proper quantization of $B'B''$, not only is the initial state of the classical game replaced by a quantum superposition, but also a probabilistic combination of the transition matrices of the classical games is replaced with a quantum superposition of the quantum multiplexers associated with each classical game.

3.4.1 A Special Case

Recall from section 3.1.3 the classical analysis of the special case of the randomized sequence of history dependent Parrondo games, with $r = (1 - r) = \frac{1}{2}$, in which one of the games is A'. The game A' has the property that regardless of history, game A is always played. Such a sequence was considered to by an instance of a history dependent game denoted by $A'B'$. In this section, a proper quantization of the randomized sequence is shown to follow as a special case of the proper quantization of the classical game $B'B''$ developed in section 3.4 above.

As before, associate the quantum multiplexer $Q' = (Q'_1, Q'_2, Q'_3, Q'_4)$, where

$$Q'_j = \sqrt{p_j}N + \sqrt{(1 - p_j)}F = \left(\begin{array}{cc} \sqrt{p_j} & -\sqrt{1 - p_j}\bar{\eta} \\ \sqrt{1 - p_j}\eta & \sqrt{p_j} \end{array} \right),$$

with the game B'. Now, first embed the game A into $SU(2)$ using basic embeddings of type 2. That is,

$$A = \sqrt{p}N^* + \sqrt{(1-p)}F^* = \begin{pmatrix} \sqrt{p}i & -\sqrt{1-p}(\overline{i\eta}) \\ \sqrt{1-p}i\eta & \sqrt{p}\bar{i} \end{pmatrix}.$$

The transition matrix for the game A' was given in Equation (3.27) and is reproduced here:

$$\Delta = \begin{pmatrix} p & 0 & p & 0 \\ 1-p & 0 & 1-p & 0 \\ 0 & p & 0 & p \\ 0 & 1-p & 0 & 1-p \end{pmatrix}.$$

The form of Δ suggests that the quantum multiplexer $Q'' = (A, A, A, A)$ should be associated with the game A'. Now let $\gamma' = \gamma'' = \frac{1}{\sqrt{2}}$in Equation (3.44) so that

$$\Sigma = \frac{1}{\sqrt{2}}(\Delta' + Q') = \frac{1}{\sqrt{2}} \begin{pmatrix} A+Q'_1 & 0 & 0 & 0 \\ 0 & A+Q'_2 & 0 & 0 \\ 0 & 0 & A+Q'_3 & 0 \\ 0 & 0 & 0 & A+Q'_4 \end{pmatrix} \qquad (3.51)$$

with

$$A + Q'_j = \begin{pmatrix} \sqrt{p}i + \sqrt{p_j} & -\left(\sqrt{1-p}(\overline{i\eta}) + \sqrt{1-p_j}\overline{\eta}\right) \\ \sqrt{1-p}i\eta + \sqrt{1-p_j}\eta & \sqrt{p}\bar{i} + \sqrt{p_j} \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{p_j} + \sqrt{p}i & -\left(\sqrt{1-p_j} - \sqrt{1-p}i\right)\overline{\eta} \\ \left(\sqrt{1-p_j} + \sqrt{1-p}i\right)\eta & \sqrt{p_j} - \sqrt{p}i \end{pmatrix}.$$

With the evaluative initial state

$$I = \frac{1}{\sqrt{\sum_{j=1}^{n} \rho_j}} \begin{pmatrix} \sqrt{\rho_1} \\ 0 \\ \sqrt{\rho_2} \\ 0 \\ \sqrt{\rho_3} \\ 0 \\ \sqrt{\rho_4} \\ 0 \end{pmatrix} \tag{3.52}$$

where the ρ_j are the probabilities that form the stationary state of the classical game $A'B'$ given in Equation (3.28), the quantum multiplexer Σ in Equation (3.44) defines a proper quantization of the classical game AB' when both A and B' are played with equal probability.

To see this, compute the output of Σ for the evaluative initial state I in Equation

(3.52):

$$\frac{1}{\sqrt{2\sum_{j=1}^{n}\rho_j}}\begin{pmatrix} \sqrt{\rho_1}(\sqrt{pi}+\sqrt{p_1}) \\ \sqrt{\rho_1}\left(\sqrt{1-p_1}+\sqrt{1-pi}\right)\eta \\ \sqrt{\rho_2}(\sqrt{pi}+\sqrt{p_2}) \\ \sqrt{\rho_2}\left(\sqrt{1-p_2}+\sqrt{1-pi}\right)\eta \\ \sqrt{\rho_3}(\sqrt{pi}+\sqrt{p_3}) \\ \sqrt{\rho_3}\left(\sqrt{1-p_3}+\sqrt{1-pi}\right)\eta \\ \sqrt{\rho_4}(\sqrt{pi}+\sqrt{p_4}) \\ \sqrt{\rho_4}\left(\sqrt{1-p_4}+\sqrt{1-pi}\right)\eta \end{pmatrix}.$$

The probability of gain produced upon measurement is

$$p^{Q}_{\text{gain}} = \frac{1}{2\sum_{j=1}^{n}\rho_j}\sum_{j=1}^{4}\left|\sqrt{\rho_j}(\sqrt{pi}+\sqrt{p_j})\right|^2 = \frac{1}{M}\sum_{j=1}^{4}\rho_j\left(\frac{p+p_j}{2}\right) = \frac{1}{M}\sum_{j=1}^{4}\rho_j q_j \tag{3.53}$$

which is exactly that given in Equation (3.29) in section 3.1.2 for the classical game $A'B'$.

3.5 A Second Proper Quantization of the Randomized Sequence of History Dependent Parrondo Games

A second proper quantization of the sequence $B'B''$ can be constructed in a manner similar to that used to construct the proper quantization for B' in section 3.3. Instead of forming a quantum superposition of the quantum multiplexers associated with each game, first embed the classical coins used in the game $B'B''$ into $SU(2)$ as

$$\begin{aligned} Y_j &= \sqrt{t_j} N + \sqrt{1-t_j} F \\ &= \begin{pmatrix} \sqrt{t_j} & -\sqrt{1-t_j}\bar{\eta} \\ \sqrt{1-t_j}\eta & \sqrt{t_j} \end{pmatrix} \end{aligned}$$

with

$$t_j = r\alpha_j + (1-r)\beta_j \quad \text{and} \quad 1-t_j = r(1-\alpha_j) + (1-r)(1-\beta_j)$$

and associate the quantum multiplexer $Y = (Y_1, Y_2, Y_3, Y_4)$ with the classical game $B'B''$. Set the initial state, as in section 3.4, equal to

$$I = \frac{1}{\sqrt{\sum_{j=1}^{n} \tau_j}} \begin{pmatrix} \sqrt{\tau_1} \\ 0 \\ \sqrt{\tau_2} \\ 0 \\ \sqrt{\tau_3} \\ 0 \\ \sqrt{\tau_4} \\ 0 \end{pmatrix}$$

where the τ_j are the probabilities that form the stationary state of the classical game $B'B''$ given in Equation (3.19). The output state of this protocol is

$$F_I = \frac{1}{\sqrt{\sum_{j=1}^{n} \tau_j}} \begin{pmatrix} \sqrt{\tau_1 t_1} \\ \sqrt{\tau_1(1-t_1)}\eta \\ \sqrt{\tau_2 t_2} \\ \sqrt{\tau_2(1-t_2)}\eta \\ \sqrt{\tau_3 t_3} \\ \sqrt{\tau_3(1-t_3)}\eta \\ \sqrt{\tau_4 t_4} \\ \sqrt{\tau_4(1-t_4)}\eta \end{pmatrix} \tag{3.54}$$

which, upon measurement produces the probability of gain

$$p_{\text{gain}}^{QB'B''} = \frac{1}{\sum_{j=1}^{n} \tau_j} \sum_{j=1}^{4} \tau_j t_j$$

which is exactly the probability of gain computed in Equation (3.29) of section 3.1.2 for the classical game AB'.

Hence, there are two approaches, both motivated by different facets of the Bleiler formalism, used here to properly quantize random sequences of Parrondo games A and B' in which each game occurs with equal probability. One approach, discussed in section 3.3, generalizes the notion of randomization between the two games via probability distributions to randomization between games via quantum superpositions. The other approach, discussed above, embeds a probabilistic combination of the games into a quantum multiplexer directly rather than via quantum superpositions of the protocols for each game.

In the former approach, note that it was crucial that game A was embedded into $SU(2)$ using basic embedding of type 2 as this allowed for the use of the broader arithmetical properties, namely factorization, of complex numbers to reproduce the classical result. In the latter on the other hand, basic embedding of type 1 sufficed.

These two different approaches to quantizing history dependent Parrondo games raise interesting questions regarding the relationship between *general* quantum history dependent Parrondo games, which are quantum multiplexers with arbitrary $SU(2)$ elements forming the diagonal blocks, and the proper quantizations of the classical history dependent Parrondo games. For example, can a general quantum history dependent Parrondo game always be factored into a sum of games which correspond to embedding of some classical history dependent Parrondo games? The reader is referred to the future directions section of chapter 6 where this subject is discussed in detail.

Chapter 4

QUANTUM LOGIC SYNTHESIS BY DECOMPOSITION

In Chapter 3, quantum multiplexers were used to properly quantize certain history dependent Parrondo games. In the following, quantum multiplexers will play a central role in synthesis of quantum logic circuits.

Recent research in generalizing quantum computation from 2-valued qubits to d-valued qudits has shown practical advantages for scaling up a quantum computer. A further generalization leads to quantum computing with *hybrid* qudits where two or more qudits have different finite dimensions. Advantages of hybrid and d-valued gates (circuits) and their physical realizations have been studied in detail by Muthukrishnan and Stroud [23], Daboul et. al [8], and Bartlett et. al [2].

Recall from section 1.1 that the evolution of state space changes the state of the qudits under the action of a unitary matrix. Because evolution matrices are viewed as quantum logic gates in quantum computing, an essential idea from the theory of classical logic circuits carries over, namely, logic synthesis. One of the goals of logic synthesis is to express a given logic gate in terms of a universal set of quantum logic gates. Recall from section 1.2.2 that sets of one and two qubit (even qudit) gates are universal.

Hence, the synthesis of a quantum logic gate requires that the corresponding matrix be decomposed to the level of unitary matrices acting on one or two qudits. Technological considerations for the implementation of one qudit gates might still require synthesis of these gates in terms of simpler one qudit rotation gates and two qudit controlled rotation gates. For 2-valued quantum computing, this is easily accomplished by the well known Euler angle parameterization of a 2×2 special unitary matrix (since a unitary matrix is equivalent to a special unitary matrix up to a complex multiple). For higher valued quantum computing, Tilma et al's work in [35] shows that a one qudit gate can be synthesized in terms of an Euler angle parametrization similar to the one available for 2×2 special unitary matrices.

If the quantum system consists of multiple qudits, then a gate may be synthesized by matrix decomposition techniques such as QR factorization and the cosine-sine Decomposition (CSD). Both the acronym CSD and the term CS decomposition will be used to refer to the cosine-sine decomposition from now on. The CSD is used by Möttönen et. al [22] and Shende et. al [31] to iteratively synthesize multi-qubit quantum circuits. Khan and Perkowski [16] use the CSD to develop an iterative synthesis method for 3-valued quantum logic circuits acting on n qudits. Bullock et. al present a synthesis method for n qudit quantum logic gates using a variation of the QR matrix factorization in [7] In [17], Khan and Perkowski give a CSD based method for synthesis of n qudit hybrid and d-valued quantum logic gates. This chapter reviews the work of these authors on quantum logic synthesis techniques based on the CS decomposition.

4.1 The Cosine-Sine Decomposition (CSD)

Let the unitary matrix $W \in \mathbf{C}^{m \times m}$ be partitioned in 2×2 block form as

$$W = \begin{array}{cc} & \begin{array}{cc} r & m-r \end{array} \\ \begin{array}{c} r \\ m-r \end{array} & \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \end{array} \tag{4.1}$$

with $2r \leq m$. Then there exist $r \times r$ unitary matrices U and X, $r \times r$ real diagonal matrices C and S, and $(m-r) \times (m-r)$ unitary matrices V and Y such that

$$W = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & I_{m-2r} \end{pmatrix} \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \tag{4.2}$$

The matrices C and S are the so-called cosine-sine matrices and are of the form C = $\mathrm{diag}(\cos\theta_1, \cos\theta_2, \ldots, \cos\theta_r)$, S = $\mathrm{diag}(\sin\theta_1, \sin\theta_2, \ldots, \sin\theta_r)$ such that $\sin^2\theta_i + \cos^2\theta_i = 1$ for some θ_i, $1 \leq i \leq r$ [34]. Algorithms for computing the CSD and the angles θ_i are given in [4, 33]. The CSD is essentially the well known singular value decomposition of a unitary matrix implemented at the block matrix level [28]. Appendix B gives a review of the CS decomposition.

The reader is advised that in the narrative that follows quantum logic gates, circuits and the corresponding unitary matrices will not be distinguished.

4.2 Synthesis of 2-valued (binary) Quantum Logic Circuits

As the authors of [16, 22, 31, 36] show, CSD gives a recursive method for synthesizing 2-valued and 3-valued n qudit quantum logic gates. In the 2-valued case the CSD of a $2^n \times 2^n$ unitary matrix W reduces to the form

$$W = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \begin{pmatrix} C & -S \\ S & C \end{pmatrix} \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \tag{4.3}$$

with each block matrix in the decomposition of size $2^{n-1} \times 2^{n-1}$.

A *quantum multiplexer* is a quantum logic gate acting on n qubits of which one is designated as the control qubit. If the control qubit of a quantum multiplexer is the lowest order qubit, that is, the first qubit in the joint state of n qubits, the multiplexer matrix is block diagonal. Note that the lowest order qubit is represented as the top most qubit in circuit diagrams. Thus, in terms of synthesis, the block diagonal matrices in Equation (4.3) are quantum multiplexers [31]. Now, depending on whether the control qubit carries $|0\rangle$ or $|1\rangle$, the gate then performs either the top left block or the bottom right block of the $n \times n$ block diagonal matrix on the remaining $(n-1)$ qubits, respectively. A circuit diagram for a n qubit quantum multiplexer with the lowest order control qubit is given in Figure 4.1 where the black circle represents control via the basis state $|1\rangle$.

Such a quantum multiplexer is expressed as

$$|a_1\rangle \otimes \begin{pmatrix} U_0 & 0 \\ 0 & U_1 \end{pmatrix} (|a_2\rangle \otimes \cdots \otimes |a_n\rangle) \tag{4.4}$$

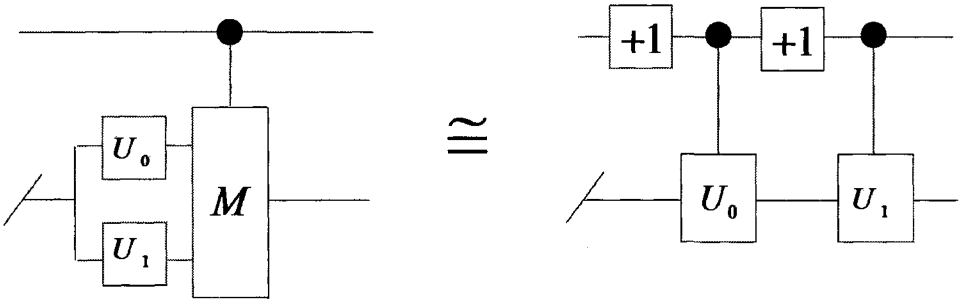


Figure 4.1: 2-valued Quantum Multiplexer M controlling the lower $(n-1)$ qubits by the top qubit. The slash symbol (/) represents $(n-1)$ qubits on the second wire. The gates labeled +1 are shifters (inverters in 2-valued logic), increasing the value of the qubit by 1 mod 2 thereby allowing for control by the highest qubit value. Depending on the value of the top qubit, one of U_t is applied to the lower qubits for $t \in \{0,1\}$.

where $|a_i\rangle$ is the i-th qubit in the circuit, and both block matrices U_0 and U_1 are of size $2^{n-1} \times 2^{n-1}$. Depending on whether $|a_1\rangle = |0\rangle$ or $|a_1\rangle = |1\rangle$, the expression (4.4) reduces to

$$|0\rangle \otimes U_0 \left(|a_2\rangle \otimes |a_3\rangle \otimes \cdots \otimes |a_n\rangle\right) \tag{4.5}$$

or

$$|1\rangle \otimes U_1 \left(|a_2\rangle \otimes |a_3\rangle \otimes \cdots \otimes |a_n\rangle\right) \tag{4.6}$$

respectively.

A *uniformly* $(n-1)$*-controlled* R_y *rotation gate* R_y is composed of a sequence of $(n-1)$-fold controlled gates $R_y^{\theta_i}$, all acting on the lowest order qubit, where

$$R_y^{\theta_i} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix}. \tag{4.7}$$

The cosine-sine matrix in Equation (4.3) is realized as a uniformly $(n-1)$-controlled

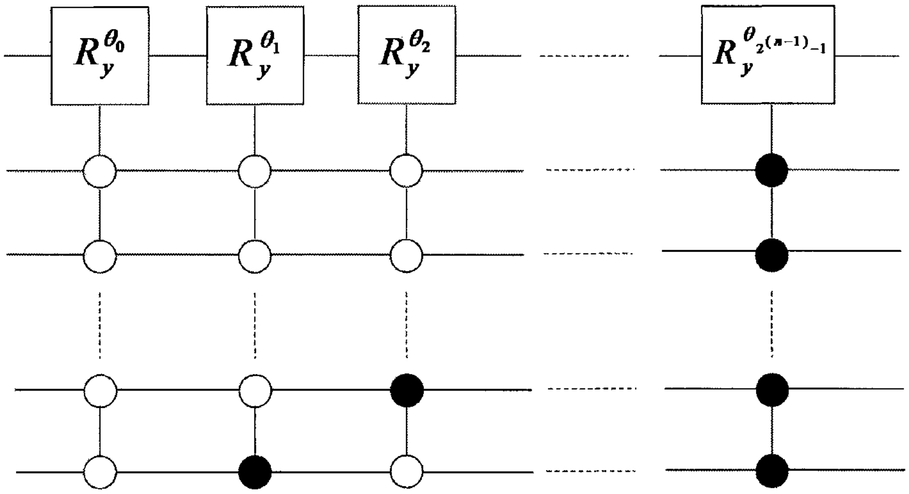


Figure 4.2: A uniformly $(n-1)$-controlled R_y rotation for 2-valued quantum logic. The ◦ control turns on for control value $|0\rangle$ and the • control turns on for control value $|1\rangle$. It requires 2^{n-1} one qubit controlled gates $R_y^{\theta_i}$ to implement a uniformly $(n-1)$-controlled R_y rotation.

R_y rotation gate, a variation of a quantum multiplexer, as shown in Figure 4.2. The control selecting the angle θ_i in the gate $R_y^{\theta_i}$ depends on which of the $(n-1)$ basis state configurations the control qubits are in at that particular stage in the circuit. In Figure 4.2, the white circle represents control via the basis state $|0\rangle$. The i-th $(n-1)$-controlled gate $R_y^{\theta_i}$ may be expressed as

$$\begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} |a_1\rangle \otimes (|a_1\rangle \otimes \cdots \otimes |a_n\rangle) \tag{4.8}$$

with θ_i taking on values from the set $\{\theta_0, \theta_1, \ldots, \theta_{2^{n-1}-1}\}$ depending on the specific configuration of $(|a_2\rangle \otimes \cdots \otimes |a_n\rangle)$, resulting in a specific $R_y^{\theta_i}$ for each i.

As an example, consider the 3 qubit uniformly 2-controlled R_y gate controlling the

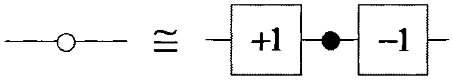


Figure 4.3: A control by input value 0 (mod 2) realized in terms of control by the highest value 1 (mod 2).

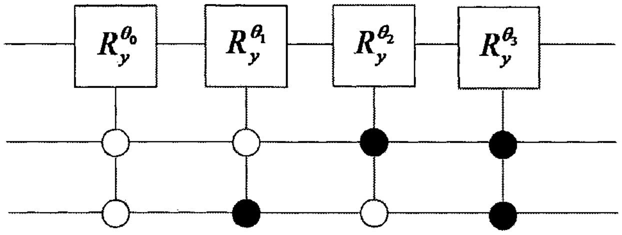


Figure 4.4: A uniformly 2-controlled R_y rotation in 2-valued logic: the lower two qubits are the control qubits and the top bit is the target bit.

top qubit from Figure 4.4. Then the action of $R_y^{\theta_i}$ on the circuit is

$$\begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} |a_1\rangle \otimes (|a_2\rangle \otimes |a_3\rangle) \tag{4.9}$$

with $\theta_i \in \{\theta_0, \theta_1, \theta_2, \theta_3\}$. As $|a_2\rangle \otimes |a_3\rangle$ takes on the values from the set $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$ in order, the expression in (4.9) reduces to the following 4 expressions respectively.

$$\begin{pmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{pmatrix} |a_1\rangle \otimes (|0\rangle \otimes |0\rangle) \tag{4.10}$$

$$\begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix} |a_1\rangle \otimes (|0\rangle \otimes |1\rangle) \tag{4.11}$$

$$\begin{pmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{pmatrix} |a_1\rangle \otimes (|1\rangle \otimes |0\rangle) \tag{4.12}$$

$$\begin{pmatrix} \cos\theta_3 & -\sin\theta_3 \\ \sin\theta_3 & \cos\theta_3 \end{pmatrix} |a_1\rangle \otimes (|1\rangle \otimes |1\rangle) \tag{4.13}$$

Observe that by iterating the CSD and factoring the result each time results in a quantum circuit consisting of variations of the quantum multiplexer.

4.3 CSD Synthesis of 3-valued (ternary) Quantum Logic Circuits

In the 3-valued case, two applications of the CSD are needed to decompose a $3^n \times 3^n$ unitary matrix W to the point where every block in the decomposition has size $3^{n-1} \times 3^{n-1}$ [16]. Choose the parameters m and r given in Equation (4.1) as $m = 3^n$ and $r = 3^{n-1}$, so that $m - r = 3^n - 3^{n-1} = 3^{n-1}(3-1) = 3^{n-1} \cdot 2$. The CS decomposition of W will now take the form in Equation (4.2), with the matrix blocks U and X of size $3^{n-1} \times 3^{n-1}$ and blocks V and Y of size $3^{n-1} \cdot 2 \times 3^{n-1} \cdot 2$. Repeating the partitioning process for the blocks V and Y with $m = 3^{n-1} \cdot 2$ and $r = 3^{n-1}$, and decomposing them with CSD followed by some matrix factoring will give rise to a decomposition of W involving unitary blocks each of size 3^{n-1} as follows.

$$W = ABC \begin{pmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & I \end{pmatrix} DEF \tag{4.14}$$

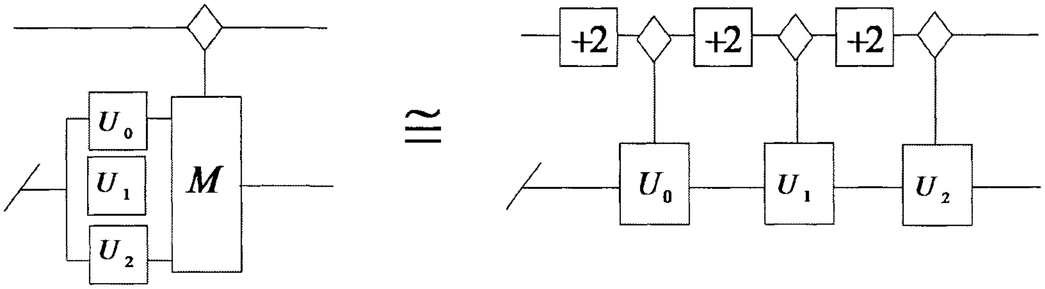


Figure 4.5: 3-valued Quantum Multiplexer M controlling the lower $(n-1)$ qutrits via the top qutrit. The slash symbol (/) represents $(n-1)$ qutrits on the second wire. The gates labeled +2 are shift gates, increasing the value of the qutrit by 2 mod 3, and the control ◇ turns on for input $|2\rangle$. Depending on the value of the top qutrit, one of U_t is applied to the lower qutrits for $t \in \{0, 1, 2\}$.

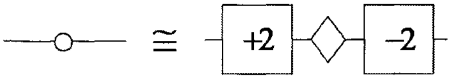


Figure 4.6: A control by the value 0 (mod 3) realized in terms of control by the highest value 2 (mod 3).

with

$$A = \begin{pmatrix} X_1 & 0 & 0 \\ 0 & X_2 & 0 \\ 0 & 0 & X_3 \end{pmatrix}, \quad B = \begin{pmatrix} I & 0 & 0 \\ 0 & C_1 & -S_1 \\ 0 & S_1 & C_1 \end{pmatrix}, \quad C = \begin{pmatrix} I & 0 & 0 \\ 0 & Z_1 & 0 \\ 0 & 0 & Z_2 \end{pmatrix} \tag{4.15}$$

$$D = \begin{pmatrix} Y_1 & 0 & 0 \\ 0 & Y_2 & 0 \\ 0 & 0 & Y_3 \end{pmatrix}, \quad E = \begin{pmatrix} I & 0 & 0 \\ 0 & C_2 & -S_2 \\ 0 & S_2 & C_2 \end{pmatrix}, \quad F = \begin{pmatrix} I & 0 & 0 \\ 0 & W_1 & 0 \\ 0 & 0 & W_2 \end{pmatrix} \tag{4.16}$$

We realize the block diagonal matrices A, C, D and F in (4.15) and (4.16) as 3-valued quantum multiplexers acting on n qutrits of which the lowest order qutrit (top

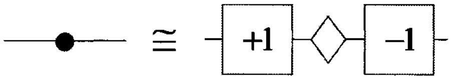


Figure 4.7: A control by the value 1 (mod 3) realized in terms of control by the highest value 2 (mod 3).

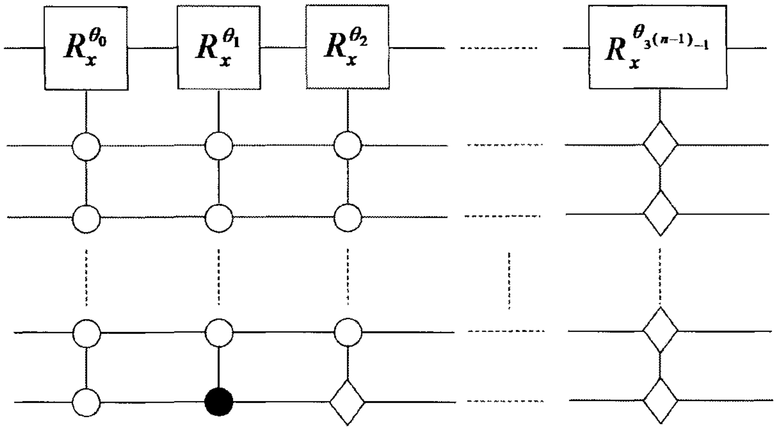


Figure 4.8: A uniformly $(n-1)$-controlled R_x rotation. The lower $(n-1)$ qutrits are the control qutrits. The controls ∘, •, and ⋄ turn on for inputs $|0\rangle$, $|1\rangle$, and $|2\rangle$ respectively. It requires 3^{n-1} one qutrit controlled gates to implement a uniformly $(n-1)$-controlled R_x or R_z rotation.

most in a circuit diagram) is designated as the control qutrit. Depending on which of the values $|0\rangle$, $|1\rangle$, or $|2\rangle$ the control qutrit carries, the gate then performs either the top left block, the middle block, or the bottom right block respectively on the remaining $n-1$ qutrits. Figure 4.5 gives the layout for a n qutrit quantum multiplexer realized in terms of *Muthukrishnan-Stroud* (MS) gates. The MS gate is a d-valued generalization of a controlled gate from 2-valued quantum logic, and allows for control of one qudit by the other via the highest value of a d-valued quantum system, which in the 3-valued case is 2 [23].

The cosine-sine matrices are realized as the uniformly $(n-1)$-controlled R_x and R_z rotations in $\mathbb{R}^3$. Similar to the 2-valued case, each R_x and R_z rotation is composed of a sequence of $(n-1)$-fold controlled gates $R_x^{\theta_i}$ or $R_z^{\phi_i}$, where

$$R_x^{\theta_i} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & -\sin\theta_i \\ 0 & \sin\theta_i & \cos\theta_i \end{pmatrix}, \quad R_z^{\phi_i} = \begin{pmatrix} \cos\phi_i & -\sin\phi_i & 0 \\ \sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.17}$$

Each $R_x^{\theta_i}$ or $R_z^{\phi_i}$ operator is applied to the top most qutrit, with the value of the angles θ_i and ϕ_i determined by the $(n-1)$ basis state configurations of the control qutrits. A uniformly controlled R_x gate is shown in Figure 4.8. Figures 4.6 and 4.7 explain the method to create controls of maximum value. Notet that the value of the control qubit is always restored in Figures 4.6 and 4.7.

4.4 Synthesis of Hybrid and d-valued Quantum Logic Circuits

It is evident from the 2 and 3-valued cases above that the CSD method of synthesis is of a general nature and can be extended to synthesis of d-valued gates acting on n qudits. In fact, it can be generalized for synthesis of hybrid n qudit gates. We propose that a $(d_1 d_2 \ldots d_n) \times (d_1 d_2 \ldots d_n)$ block diagonal unitary matrix be regarded as a quantum multiplexer for an n qudit hybrid quantum state space $\mathcal{H} = \mathcal{H}_{d_1} \otimes \mathcal{H}_{d_2} \otimes \cdots \otimes \mathcal{H}_{d_n}$, where $\mathcal{H}_{d_i}$ is the state space of the i qudit.

Moreover, consider a cosine-sine matrix of size $(d_1 d_2 \ldots d_n) \times (d_1 d_2 \ldots d_n)$ of the form

$$\begin{pmatrix} I_p & 0 & 0 & 0 \\ 0 & C & -S & 0 \\ 0 & S & C & 0 \\ 0 & 0 & 0 & I_q \end{pmatrix} \tag{4.18}$$

with I_p and I_q both some appropriate sized identity matrices, $C = \text{diag}(\cos\theta_1, \cos\theta_2, \ldots, \cos\theta_t)$ and $S = \text{diag}(\sin\theta_1, \sin\theta_2, \ldots, \sin\theta_t)$ such that $\sin^2\theta_i + \cos^2\theta_i = 1$ for some θ_i with $1 \leq i \leq t$, and $p + q + 2t = (d_1 d_2 \ldots d_n)$. We regard this matrix as a *uniformly controlled Givens rotation* matrix, a generalization of the R_y, R_x, and R_z rotations of the 2 and 3-valued cases. A Givens rotation matrix has the general form

$$G^{\theta}_{(i,j)} = \begin{pmatrix} 1 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \ldots & \cos\theta & \ldots & -\sin\theta & \ldots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \ldots & \sin\theta & \ldots & \cos\theta & \ldots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \ldots & 0 & \ldots & 1 \end{pmatrix} \tag{4.19}$$

where the cosine and sine values reside in the intersection of the i-th and j-th rows and columns, and all other diagonal entries are 1 [12]. Hence, a Givens rotation matrix corresponds to a rotation by some angle θ in the ij-th hyperplane.

Based on the preceding discussion, we give in Theorem 4.4.1 below an iterative CSD method for synthesizing a n qudit hybrid quantum circuit by decomposing the corresponding unitary matrix of size $(d_1d_2 \ldots d_n) \times (d_1d_2 \ldots d_n)$ in terms of quantum multiplexers and uniformly controlled Givens rotations. As a consequence of Theorem 4.4.1, we give in corollary 4.4.1 a CSD synthesis of a quantum quantum logic circuit with corresponding unitary matrix of size $d^n \times d^n$. The synthesis methods given above for 2-valued and 3-valued circuits may then be treated as special cases of the former.

4.4.1 Hybrid Quantum Logic Circuits

Consider a hybrid quantum state space of a n qudits, $\mathcal{H} = \mathcal{H}_{d_1} \otimes \mathcal{H}_{d_2} \otimes \cdots \otimes \mathcal{H}_{d_n}$, where each qudit may be of distinct d-valued dimension d_i, $1 \leq i \leq n$. Since a qudit in $\mathcal{H}$ is a column vector of length $d_1d_2 \ldots d_n$, a quantum logic gate acting on such a vector is a $(d_1d_2 \ldots d_N) \times (d_1d_2 \ldots d_n)$ unitary matrix W. We will decompose W, using CSD iteratively, from the level of n qudits to $(n-1)$ qudits in terms of quantum multiplexers and uniformly controlled Givens rotations. However, since the d-valued dimension may be different for each qudit, the block matrices resulting from the CS decomposition may not be of the form $d^{n-1} \times d^{n-1}$ for some d. Therefore, we proceed by choosing one of the qudits, c_{d_i} of dimension d_i, to be the control qudit and order of the basis of $\mathcal{H}$ in such a way that c_{d_i} is the highest order qudit. We will decompose W with respect to c_{d_i} so that the resulting quantum multiplexers are controlled by c_{d_i} and the uniformly controlled Givens rotations control c_{d_i} via the remaining $(n-1)$ qudits. We give the synthesis method in Theorem 4.4.1 below.

Theorem 4.4.1. Let W be an $M \times M$ unitary matrix, with $M = d_1d_2 \ldots d_n$, acting as a quantum logic gate on a quantum hybrid state space $\mathcal{H} = \mathcal{H}_{d_1} \otimes \mathcal{H}_{d_2} \otimes \cdots \otimes \mathcal{H}_{d_n}$ of n qudits. Then W can be synthesized with respect to a control qudit c_{d_i} of dimension

d_i, having the highest order in $\mathcal{H}$, iteratively from level n to level $(n-1)$ in terms of quantum multiplexers and uniformly controlled Givens rotations.

Proof. **Step 1.** At level n, identify a control qudit c_{d_i} of dimension d_i. Reorder the basis of $\mathcal{H}$ so that c_{d_i} is the highest order qudit and the new state space isomorphic to $\mathcal{H}$ is $\bar{\mathcal{H}} = \mathcal{H}_{d_i} \otimes \mathcal{H}_{d_2} \otimes \cdots \otimes \mathcal{H}_{d_1} \otimes \cdots \otimes \mathcal{H}_{d_n}$.

If we choose values for the CSD parameters m and r as $m = (d_1 d_2 \ldots d_n)$ and $r = (d_1 d_2 \ldots d_{i-1} d_{i+1} \ldots d_n)$, then $m - r = d_1 \ldots d_{i-1} d_{i+1} \ldots d_n(d_i - 1)$. Decomposing W by CSD, we get the form in (4.2) with the matrix blocks U and X of size $r \times r$ and blocks V and Y of size $(m-r) \times (m-r)$. Should $m - r$ not have the factor $(d_i - 1)$, we would achieve the desired decomposition of W from level of n qudits to the level of $(n-1)$ qudits in terms of block matrices of size $r \times r$. The task therefore is to divide out the factor $(d_i - 1)$ from $m - r$ by an iterative *lateral decomposition* described below, that uses the CSD to cancel $(d_i - 1)$ from $m - r$ at each iteration level leaving only blocks of size $r \times r$.

For step 2 of the proof below, we will say that a matrix with k rows and k columns has size k instead of $k \times k$.

Step 2. *Iterative Lateral Decomposition*: For the unitary matrix W of size M, we define the j-th lateral decomposition of W as the CS decomposition of all block matrices of size other than r that result from the $(j-1)$-st lateral decomposition of W:

For $0 \leq j \leq (d_i - 2)$, *set*

$m_0 = (d_1 d_2 \ldots d_n)$

$r_0 = (d_1 d_2 \ldots d_{i-1} d_{i+1} \ldots d_n)$

If $j = 0$

```
Apply CSD to W
Else set
$m_j = m_0 - j \cdot r_0$
$r_j = r_0$
$m_j - r_j = m_0 - (j+1)r_0$
$=(d_1 d_2 \ldots d_{i-1} d_{i+1} \ldots d_n)\left[d_i - (j+1)\right]$
$m_j - 2r_j = m_0 - (j+2)r_0$
$=(d_1 d_2 \ldots d_{i-1} d_{i+1} \ldots d_n)\left[d_i - (j+2)\right]$
Apply CSD to matrix blocks of size other than $r_0$ from step $j-1$
End If
End For.
```

When $j = 0$, we call the resulting 0-th lateral decomposition the *global decomposition*. Note that if $d_i = 2$, then the algorithm for the lateral decomposition stops after the global decomposition. This suggests that whenever feasible, the control system in the quantum circuit should be 2-valued so as to reduce the number of iterations . Below we give a matrix description of the algorithm.

For $j = 0$, the 0-th lateral decomposition of W will just be the CS decomposition of W.

$$W = A_0^{(0)} B_0^{(0)} D_0^{(0)} \tag{4.20}$$

where

$$A_0^{(0)} = \begin{pmatrix} U_0^{(0)} & 0 \\ 0 & V_0^{(0)} \end{pmatrix}, B_0^{(0)} = \begin{pmatrix} C_0^{(0)} & -S_0^{(0)} & 0 \\ S_0^{(0)} & C_0^{(0)} & 0 \\ 0 & 0 & I_{m_0-2r_0} \end{pmatrix} D_0^{(0)} = \begin{pmatrix} X_0^{(0)} & 0 \\ 0 & Y_0^{(0)} \end{pmatrix}$$

with $U_0^{(0)}$, $X_0^{(0)}$, $C_0^{(0)}$, and $S_0^{(0)}$ all of the desired size r_0, while $V_0^{(0)}$ and $Y_0^{(0)}$ are of size $m_0 - r_0$. The superscripts label the iteration step, in this case $j = 0$. The subscript is used to distinguish between the various matrix blocks U, V, X, Y, C, S, that occur at the various levels of iteration. The 0-th lateral decomposition in the form from Equation (4.20) is called the *global decomposition* of W.

For $j = 1$, we perform lateral decomposition on the blocks $V_0^{(0)}$ and $Y_0^{(0)}$ of the block matrices $A_0^{(0)}$ and $D_0^{(0)}$ respectively, the only blocks of size other than r_0 resulting from the 0-th lateral decomposition given in (4.20). In both cases, set $m_1 = m_0 - r_0$ and $r_1 = r_0$ so that $m_1 - r_1 = m_0 - 2r_0$. For $V_0^{(0)}$ this gives the decomposition

$$A_0^{(0)} = \begin{pmatrix} U_0^{(0)} & 0 \\ 0 & \begin{pmatrix} U_0^{(1)} & 0 \\ 0 & V_0^{(1)} \end{pmatrix} \begin{pmatrix} C_0^{(1)} & -S_0^{(1)} & 0 \\ S_0^{(1)} & C_0^{(1)} & 0 \\ 0 & 0 & I_{m_0-3r_0} \end{pmatrix} \begin{pmatrix} X_0^{(1)} & 0 \\ 0 & Y_0^{(1)} \end{pmatrix} \end{pmatrix} \tag{4.21}$$

with $U_0^{(1)}, X_0^{(1)}, C_0^{(1)}$ and $S_0^{(1)}$ all of size r_0, and $V_0^{(1)}$ and $Y_0^{(1)}$ of size $m_1 - r_1$. All three matrices residing in the lower block diagonal of the matrix (4.21) are the same size. Therefore, by introducing identity matrices of size r_0 and factoring out at the matrix block level, $A_0^{(0)}$ will be updated to

$$A_0^{(0)} = A_0^{(1)} B_0^{(1)} D_0^{(1)} \tag{4.22}$$

where

$$A_0^{(1)} = \begin{pmatrix} U_0^{(0)} & 0 & 0 \\ 0 & U_0^{(1)} & 0 \\ 0 & 0 & V_0^{(1)} \end{pmatrix}, B_0^{(1)} = \begin{pmatrix} I_{r_0} & 0 & 0 & 0 \\ 0 & C_0^{(1)} & -S_0^{(1)} & 0 \\ 0 & S_0^{(1)} & C_0^{(1)} & 0 \\ 0 & 0 & 0 & I_{m_0-3r_0} \end{pmatrix},$$

$$D_0^{(1)} = \begin{pmatrix} I_{r_0} & 0 & 0 \\ 0 & X_0^{(1)} & 0 \\ 0 & 0 & Y_0^{(1)} \end{pmatrix}$$

A similar lateral decomposition of the block $Y_0^{(0)}$ will update $D_0^{(0)}$ in (4.20) to

$$C_0^{(0)} = A_1^{(1)} B_1^{(1)} D_1^{(1)} \tag{4.23}$$

where

$$A_1^{(1)} = \begin{pmatrix} X_0^{(0)} & 0 & 0 \\ 0 & U_1^{(1)} & 0 \\ 0 & 0 & V_1^{(1)} \end{pmatrix}, B_1^{(1)} = \begin{pmatrix} I_{r_0} & 0 & 0 & 0 \\ 0 & C_1^{(1)} & -S_1^{(1)} & 0 \\ 0 & S_1^{(1)} & C_1^{(1)} & 0 \\ 0 & 0 & 0 & I_{m_0-3r_0} \end{pmatrix},$$

$$D_1^{(1)} = \begin{pmatrix} I_{r_0} & 0 & 0 \\ 0 & X_1^{(1)} & 0 \\ 0 & 0 & Y_1^{(1)} \end{pmatrix}$$

For iteration $j \neq 0$, perform lateral decomposition on the total 2^j blocks $V_k^{(j-1)}$, $Y_k^{(j-1)}$, where $0 \leq k \leq 2^{(j-1)} - 1$, that occur in the global decomposition at the end of iteration $(j-1)$. For each $V_k^{(j-1)}$, $Y_k^{(j-1)}$, set $r_j = r_0$, $m_j = m_{j-1} - r_{j-1} = m_0 - jr_0$. For each $V_k^{(j-1)}$, the lateral decomposition at level j will give the following

$$A_k^{(j-1)} = \begin{pmatrix} \Delta^{(j-1)} & 0 \\ 0 & \begin{pmatrix} U_{k'}^{(j)} & 0 \\ 0 & V_{k'}^{(j)} \end{pmatrix} \begin{pmatrix} C_{k'}^{(j)} & -S_{k'}^{(j)} & 0 \\ S_{k'}^{(j)} & C_{k'}^{(j)} & 0 \\ 0 & 0 & I_{m_0-(j+2)r_0} \end{pmatrix} \begin{pmatrix} X_{k'}^{(j)} & 0 \\ 0 & Y_{k'}^{(j)} \end{pmatrix} \end{pmatrix} \tag{4.24}$$

where the $\Delta^{(j-1)}$ is the block diagonal matrix of size of $j \cdot r_0$ arising from the lateral decomposition in the previous j steps. The blocks $U_{k'}^{(j)}$, $X_{k'}^{(j)}$, $C_{k'}^{(j)}$ and $S_{k'}^{(j)}$ are all of size r_0, $0 \leq k' \leq 2^j - 1$. The blocks $V_{k'}^{(j)}$ and $Y_{k'}^{(j)}$ are of size $m_j - r_j$. The three matrices residing in the lower block diagonal of the matrix (4.24) are all of same size. Therefore, by introducing identity matrices of size $j \cdot r_0$ and factoring out at the block level, $A_k^{(j-1)}$ will be updated to

$$A_k^{(j-1)} = A_{k'}^{(j)} B_{k'}^{(j)} D_{k'}^{(j)}$$

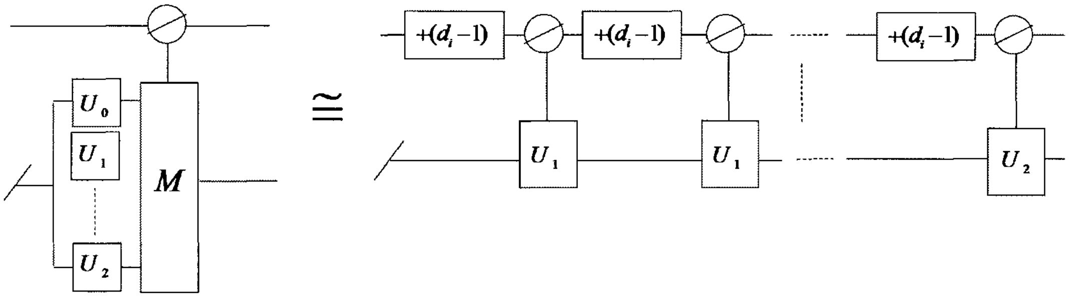


Figure 4.9: An n qudit hybrid quantum multiplexer, here realized in terms of Muthukrishnan-Stroud (d-valued controlled) gates. The top qudit has dimension d_i and controls the remaining $(n-1)$ qudits of possibly distinct dimensions which are represented here by the symbol (/). The control ⊘ turns on for input value $|d_i - 1\rangle$ mod d_i of the controlling signal coming from the top qudit.The gates $+(d_i - 1)$ shift the values of control qudit by $(d_i - 1)$ mod d_i.

where

$$A_k^{(j-1)} = \begin{pmatrix} \Delta^{(j-1)} & 0 & 0 \\ 0 & U_{k'}^{(j)} & 0 \\ 0 & 0 & V_{k'}^{(j)} \end{pmatrix}, B_{k'}^{(j)} = \begin{pmatrix} I_{j \cdot r_0} & 0 & 0 & 0 \\ 0 & C_{k'}^{(j)} & -S_{k'}^{(j)} & 0 \\ 0 & S_{k'}^{(j)} & C_{k'}^{(j)} & 0 \\ 0 & 0 & 0 & I_{m_0-(j+2)r_0} \end{pmatrix}$$

$$D_{k'}^{(j)} = \begin{pmatrix} I_{j \cdot r_0} & 0 & 0 \\ 0 & X_{k'}^{(j)} & 0 \\ 0 & 0 & Y_{k'}^{(j)} \end{pmatrix}$$

For the next iteration, set $k = k'$ and iterate. Upon completion of the lateral decomposition, repeat steps 1 and 2 for the synthesis of the circuit for the remaining $(n-1)$ qudits, with the restriction that each gate in the remaining circuit be decomposed with respect to the same control qudit identified in step 1. □

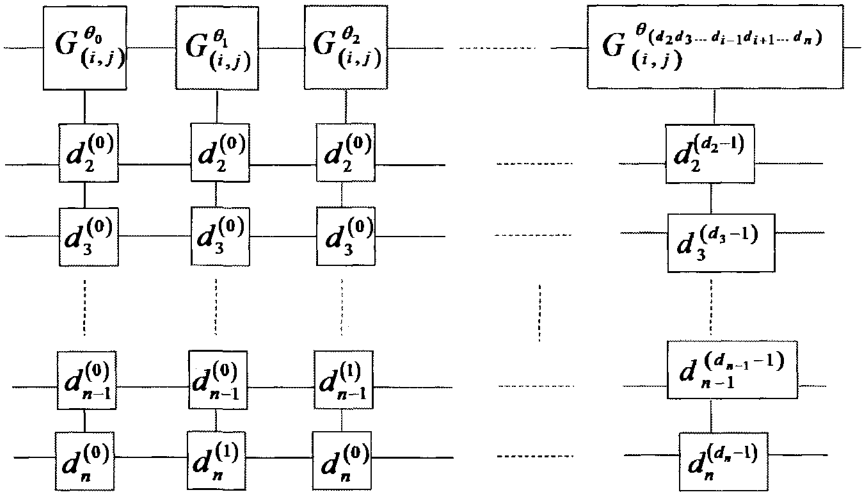


Figure 4.10: A hybrid uniformly $(n-1)$-controlled Givens rotation. The lower $(n-1)$ qudits of dimensions $d_2, d_3, \dots, d_{i-1}, d_{i+1}, \dots, d_n$, respectively, are the control qudits, and the top is the target qudit of dimension d_i. The control gate $d_l^{(k)}$ turns on whenever the control qudit of dimension d_l takes on the value k (mod) d_l.

Since the basis for $\mathcal{H}$ was reordered in the beginning so that the control qudit was of the highest order, the block diagonal matrices with all blocks of size $r_0 \times r_0$ are interpreted as quantum multiplexers and the cosine-sine matrices are interpreted as uniformly controlled Givens rotations. In Figures 4.9 and 4.10, we present the circuit diagrams of a hybrid quantum multiplexer and a uniformly controlled Givens rotation, respectively. A uniformly controlled Givens rotation matrix on n qudits can be realized as the composition of various $(n-1)$-fold controlled Givens rotation matrices, $G^{\theta_k}_{(i,j)}$, acting on the top most qudit of the circuit with the angle of rotation depending on the basis state configuration, in their respective dimensions, of the lower $(n-1)$ qudits.

4.4.2 d-valued Quantum Logic Circuits

Given the hybrid n qudit synthesis, the case of d-valued synthesis becomes a special case of the former since by setting all $d_i = d$, the state space $\mathcal{H} = \mathcal{H}_{d_1} \otimes \mathcal{H}_{d_2} \otimes \cdots \otimes \mathcal{H}_{d_n}$ reduces to the state space $H_d^{\otimes n}$. Unitary operators acting on the states in $H_d^{\otimes n}$ are unitary matrices of size $d^n \times d^n$. We give the following result for d-valued synthesis.

Corollary 4.4.1. A d-valued n qudit quantum logic gate can be synthesized in terms of quantum multiplexers and uniformly controlled Givens rotations.

Proof: Since all the qudits are of the same dimension, there is no need to choose a control qudit. In the proof of Theorem 4.4.1, set $d_i = d$ for all i. Then $M = d_1 d_2 \ldots d_n = d^n$. For iteration $j = 0$ of the lateral decomposition, set $m_0 = d^n$, $r_0 = d^{n-1}$, so that $m_0 - r_0 = d^{n-1}(d-1)$. For $0 \leq j \leq (d-2)$, set $r_j = r_0 = d^{n-1}$, and $m_j = m_{j-1} - r_{j-1} = d^{n-1}(d-(j+1))$.

For the d-valued case, we note that there are a total of $d^{n-1}(2^{d-1}-1)$ one qudit Givens rotations in the circuit at the $(n-1)$ level, each arising from the $\sum_{i=0}^{(d-2)} 2^i = 2^{d-1} - 1$ uniformly controlled Givens rotations in the CS decomposition of an n qudit gate. Moreover, in each uniformly controlled Givens rotation, there are $(n-1)d^{n-1}$ control symbols of which $(n-1)d^{n-2}$ correspond to control by the highest value of $d-1$. The latter controls do not require shift gates around them to increase the value of the signal qudit to $d-1$. Hence, there are $(n-1)d^{n-1} - (n-1)d^{n-2} = (n-1)(d^{n-1} - d^{n-2})$ control symbols that correspond to control by values other than $d-1$ and therefore need two shift gates (fig. 11) around them. This gives the total number of one qudit shift gates in each uniformly controlled rotation to be $2(n-1)(d^{n-1} -$

$d^{n-2})$, whereby the total number of one qudit shifts and Givens rotations in the circuit at the $(n-1)$ level is $2(n-1)(d^{n-1}-d^{n-2})(2^{d-1}-1)+d^{n-1}(2^{d-1}-1)=(2^{d-1}-1)\,[2(n-1)(d^{n-1}-d^{n-2})+d^{n-1}]$.

There are 2^{d-1} quantum multiplexers in the decomposition, each consisting of a total of $2d$ shift and controlled gates. Hence, there are a total of $d\cdot 2^{d}$ one qudit and controlled gates in the $(n-1)$ level circuit. This gives a total, worst case, one qudit and controlled gate count in the circuit at level $(n-1)$ to be $(2^{d-1}-1)\,[2(n-1)(d^{n-1}-d^{n-2})+d^{n-1}]+d\cdot 2^{d}$.

Chapter 5

A QUATERNIONIC CO-ORDINATIZATION OF BINARY QUANTUM COMPUTATION

A quaternionic coordinatization of the players' quantum strategies in certain quantized games by Landsburg in [18] gives him a computational framework for classifying potential Nash equilibria in these games. This idea led Ahmed, Bleiler and Khan [1] to construct a parallel coordinatization using octonions for another class of quantized games, giving the authors a computational framework for classifying potential Nash equilibria in these games. Motivated by these result, this chapter proposes a quaternionic coordinatization of binary quantum computation by putting quaternionic coordinates on the Lie group $SU(2)$ of quantum logic gates acting on one qubit and on the projective complex state space $\mathbb{C}P^1$ of one qubit, with the eventual goal of providing an enhanced computational capability for circuit analysis.

In general, one qubit quantum logic gates are unitary matrices with determinant 1 or -1. However, a 2×2 unitary matrix is equivalent to a special unitary matrix up to a factor of $\bar{i}$. That is, if

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is a unitary matrix with determinant $ad - cb = -1$, then

$$U = \bar{i}U' = \bar{i}\begin{pmatrix} ia & ib \\ ic & id \end{pmatrix}$$

where U' has determinant $(ia)(id)-(ic)(ib) = -ad+cb = -1(ad-cb) = (-1)(-1) = 1$ and is therefore special unitary. The factor $\bar{i}$ is regarded a unitary phase in any resulting calculations. For the remainder of this chapter, all instances of a unitary matrix U with determinant -1 will be replaced with its equivalent special unitary matrix $U' \in SU(2)$.

Now by identifying both $SU(2)$, the set of one qubit quantum logic gates, and $\mathbb{C}P^1$, the state space of a qubit, with unit quaternions $Sp(1)$, we develop a quaternionic co-ordinatization of binary quantum computation. In this chapter, we will use the notation $\mathbb{1}$ and $\mathbb{J}$ for the unit quaternions 1 and j respectively to emphasize their roles as control signals in the context of quantum computing.

5.1 Identifying $SU(2)$ with $Sp(1)$

The Lie group $Sp(1)$ of unit quaternions can be considered as

$$Sp(1) = \left\{u = u_0\mathbb{1} + u_1\mathbb{J} : |u|^2 = |u_0|^2 + |u_1|^2 = (u_0')^2 + (u_1')^2 + (u_0'')^2 + (u_1'')^2 = 1\right\}.$$

The Lie group $SU(2)$ of 2×2 special unitary matrices is

$$SU(2) = \left\{\begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix} : \alpha, \beta \in \mathbb{C} \text{ and } |\alpha|^2 + |\beta|^2 = 1\right\}$$

The special unitary requirement suggests a strong connection between $SU(2)$ and $Sp(1)$. Indeed, we can set up a one to one correspondence between $SU(2)$ and $Sp(1)$ as follows. Consider $\mathbb{H}$ as $\mathbb{C}^2$ under the identification

$$\alpha\mathbb{1} + \beta\mathbb{J} \longleftrightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

and let $y_1\mathbb{1} + y_2\mathbb{J} \in \mathbb{H}$ and $z_1\mathbb{1} + z_2\mathbb{J} \in Sp(1)$. Recall that $\alpha\mathbb{J} = \mathbb{J}\overline{\alpha}$ for all $\alpha \in \mathbb{C}$ and form the product

$$\begin{aligned}(y_1\mathbb{1} + y_2\mathbb{J})(z_1\mathbb{1} + z_2\mathbb{J}) &= y_1z_1\mathbb{1} + y_2\mathbb{J}z_1 + y_1z_2\mathbb{J} + y_2\mathbb{J}z_2\mathbb{J} \\ &= (y_1z_1 - y_2\overline{z}_2)\mathbb{1} + (y_2\overline{z}_1 + y_1z_2)\mathbb{J}.\end{aligned}$$

Write this result as an element of $\mathbb{C}^2$ via the identification as

$$\begin{pmatrix} y_1z_1 - y_2\overline{z}_2 \\ y_2\overline{z}_1 + y_1z_2 \end{pmatrix}.$$

But

$$\begin{pmatrix} y_1z_1 - y_2\overline{z}_2 \\ y_2\overline{z}_1 + y_1z_2 \end{pmatrix} = \begin{pmatrix} z_1 & -\overline{z}_2 \\ z_2 & \overline{z}_1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

so the result of the quaternionic product, as an element of $\mathbb{C}^2$, is in the image of the special unitary transformation

$$\begin{pmatrix} z_1 & -\overline{z}_2 \\ z_2 & \overline{z}_1 \end{pmatrix}$$

acting on $\mathbb{C}^2$, establishing the following identification of $SU(2)$ and $Sp(1)$

$$\begin{pmatrix} z_1 & -\overline{z}_2 \\ z_2 & \overline{z}_1 \end{pmatrix} \longleftrightarrow z_1\mathbb{1} + z_2\mathbb{J}.$$

In other words, *right* multiplication by a unit quaternion in $\mathbb{H}$ corresponds to the action *on the left* of the corresponding special unitary matrix on $\mathbb{C}^2$.

In fact, it is possible to make the quaternionic product compatible with the left action of a $SU(2)$ element on $\mathbb{C}^2$. That is, the left action of a linear transformation on $\mathbb{C}^2$ can be made to correspond to multiplication on the *left* by a unit quaternion in $\mathbb{H}$ by writing quaternions with scalars *on the right.* For then, we get

$$\begin{aligned}(\mathbb{1}z_1 + \mathbb{J}z_2)(\mathbb{1}y_1 + \mathbb{J}y_2) &= \mathbb{1}z_1y_1 + \mathbb{J}z_2y_1 + z_1\mathbb{J}y_2 + \mathbb{J}z_2\mathbb{J}y_2 \\ &= \mathbb{1}(z_1y_1 - \overline{z}_2y_2) + \mathbb{J}(z_2y_1 + z_1\overline{y}_2)\end{aligned}$$

which corresponds to

$$\begin{pmatrix} z_1y_1 - \overline{z}_2y_2 \\ z_2y_1 + z_1\overline{y}_2 \end{pmatrix} = \begin{pmatrix} z_1 & -\overline{z}_2 \\ z_2 & \overline{z}_1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

with the identification of $SU(2)$ and $Sp(1)$ given by

$$\begin{pmatrix} z_1 & -\overline{z}_2 \\ z_2 & \overline{z}_1 \end{pmatrix} \longleftrightarrow \mathbb{1}z_1 + \mathbb{J}z_2.$$

Either of these two identifications of $SU(2)$ with $Sp(1)$ introduces quaternionic co-ordinates on $SU(2)$. We choose the latter due its salient property of keeping the quaternionic product compatible with the left action of $SU(2)$ on $\mathbb{C}^2$. In other words, we consider the quaternions as a right complex vector space.

It is an easy check that this identification preserves multiplication in $SU(2)$. If

$$\begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix}, \begin{pmatrix} \delta & -\overline{\gamma} \\ \gamma & \overline{\delta} \end{pmatrix} \in SU(2),$$

then their product in $SU(2)$ results in

$$\begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix} \begin{pmatrix} \delta & -\overline{\gamma} \\ \gamma & \overline{\delta} \end{pmatrix} = \begin{pmatrix} \alpha\delta - \overline{\beta}\gamma & -\alpha\overline{\gamma} - \overline{\beta\delta} \\ \beta\delta + \overline{\alpha}\gamma & -\beta\overline{\gamma} + \overline{\alpha\delta} \end{pmatrix}$$

which is identified with the unit quaternion

$$\mathbb{1}(\alpha\delta - \overline{\beta}\gamma) + \mathbb{J}(\beta\delta + \overline{\alpha}\gamma), \qquad (5.1)$$

while identifying the $SU(2)$ elements with unit quaternions *first* results in the quaternionic product

$$(\mathbb{1}\alpha + \mathbb{J}\beta)(\mathbb{1}\delta + \mathbb{J}\gamma) = \mathbb{1}(\alpha\delta - \overline{\beta}\gamma) + \mathbb{J}(\beta\delta + \overline{\alpha}\gamma)$$

the result of which is exactly the quaternion in (5.1). In fact, this identification sets up a Lie group isomorphism between $Sp(1)$ and $SU(2)$.

5.2 Identifying $Sp(1)$ with $\mathbb{C}P^1$

Observe that the Bloch sphere

$$\mathbb{C}P^1 \equiv \left(\mathbb{C}^2 - \{0\}\right)/\mathbb{C}^* \cong S^3/U(1)$$

where $\mathbb{C}^* = \mathbb{R}^+ \times U(1)$ and

$$\begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} x\lambda \\ y\lambda \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \lambda$$

for $x, y \in \mathbb{C}$, both not equal to zero, and the scalar $\lambda \in U(1)$ and is called *phase.* Note that we scalar multiply elements of $\mathbb{C}P^1$ on the *right* rather than the left, a convention that is necessary for differentiating between scalar multiplication and the action of $SU(2)$ on $\mathbb{C}P^1$ under the identifications.

The Hopf map $H : S^3 \to \mathbb{C}P^1$ is defined here as

$$H : \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto yx^{-1}$$

with 0^{-1} considered to be the number $\frac{1}{0}$. On the Bloch sphere, the pure states are represented by $\frac{0}{1}$ and $\frac{1}{0}$ corresponding to the vectors

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

respectively. In general, $\frac{\beta}{\alpha} = \beta\alpha^{-1}$ corresponds to the vector

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

and up to unitary phase

$$\begin{pmatrix} \alpha\lambda \\ \beta\lambda \end{pmatrix} \equiv \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

We identify this element of the Bloch sphere with a unit quaternion representing its orbit in S^3. That is,

$$\begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \mathbb{1}x + \mathbb{J}y \tag{5.2}$$

where

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \longmapsto \mathbb{1}, \begin{pmatrix} i \\ 0 \end{pmatrix} \longmapsto \mathbb{I}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \longmapsto \mathbb{J}, \begin{pmatrix} 0 \\ -i \end{pmatrix} \longmapsto \mathbb{K} \tag{5.3}$$

is the identification of the basis of $\mathbb{C}^2$ (hence $\mathbb{C}P^1$) with the basis of $\mathbb{H}$ as complex vector spaces. The identifications in equations (5.2) and (5.3) induce a product between elements of $SU(2)$ and elements of $\mathbb{C}P^1$ via quaternionic multiplication that is consistent with the left action of an appropriate $SU(2)$ element on the elements of $\mathbb{C}P^1$. That is, for

$$A = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \Delta = \begin{pmatrix} \delta \\ \gamma \end{pmatrix} \in \mathbb{C}P^1, \begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix} \in SU(2)$$

quaternionic multiplication gives the product $\star$ between A and Δ as follows.

$$\begin{aligned} A \star \Delta = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \star \begin{pmatrix} \delta \\ \gamma \end{pmatrix} &= (\mathbb{1}\alpha + \mathbb{J}\beta)(\mathbb{1}\delta + \mathbb{J}\gamma) \\ &= \mathbb{1}(\alpha\delta - \overline{\beta}\gamma) + \mathbb{J}(\beta\delta + \overline{\alpha}\gamma) \\ &= \begin{pmatrix} \alpha\delta - \overline{\beta}\gamma \\ \beta\delta + \overline{\alpha}\gamma \end{pmatrix} = \begin{pmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{pmatrix} \begin{pmatrix} \delta \\ \gamma \end{pmatrix} \end{aligned}$$

5.2.1 Action of $U(1)$ on $\mathbb{C}P^1$

Note that the unit complex numbers $U(1)$ can be embedded into $SU(2)$ via

$$\alpha \hookrightarrow \begin{pmatrix} \alpha & 0 \\ 0 & \overline{\alpha} \end{pmatrix}$$

and in this form act on $\mathbb{C}P^1$ as linear transformation instead of scalar multiplication. Our identifications respect this fact, as the following example shows.

$$\begin{aligned} \begin{pmatrix} \alpha & 0 \\ 0 & \overline{\alpha} \end{pmatrix} \begin{pmatrix} \delta \\ \gamma \end{pmatrix} = \begin{pmatrix} \alpha\delta \\ \overline{\alpha}\gamma \end{pmatrix} \longmapsto \mathbb{1}(\alpha\delta) + \mathbb{J}(\overline{\alpha}\gamma) &= \alpha\mathbb{1}\delta + \alpha\mathbb{J}\gamma \\ &= \alpha(\mathbb{1}\delta + \mathbb{J}\gamma). \end{aligned}$$

Note that even though in the expression $\alpha(\mathbb{1}\delta + \alpha\mathbb{J}\gamma)$ the complex number α appears on the left, it does not represent scalar multiplication because of our convention that scalars multiply on the right. In fact, it's occurrence on the left of the quaternion $\mathbb{1}\delta + \alpha\mathbb{J}\gamma$ tells us that it represents the action of $U(1)$ as a linear transformation under the embedding in $SU(2)$.

Chapter 6

FUTURE DIRECTIONS

The proper quantization protocols developed in chapter 3 for history dependent Parrondo games using certain quantum multiplexers lend a game theoretic perspective to the study of quantum logic circuits via quantum multiplexers. Indeed, the notion of the Parrondo effect is now attached to quantum circuits, and it is natural to raise the following question: 1) can a genuine "quantum Parrondo effect" be characterized in quantum circuits through this game theoretic perspective?

Moreover, to date there is no agreement in the literature on exactly what a quantum Markov process is. One difficulty lies in coming up with an appropriate definition of the "quantum" stable state. Our quantizations of history dependent Parrondo games are essentially specific quantized Markov processes involving specific elements of the Lie group $SU(2)$ and with stable states chosen game-theoretically. A more general set up is possible in which arbitrary elements of $SU(2)$ are utilized. In such a set up, is it possible to use quantum game theory to come up with a natural choice for the stable state? Moreover, is it possible to characterize a quantized version of the Parrondo effect in this general set up, and if so, what does it mean for quantum computation?

To be more precise, the work in Chapter 3 embeds classical history dependent Parrondo games into quantum multiplexers via embeddings of type 1 and 2. The resulting quantum multiplexers, when made to act upon a particular evaluative initial state, reproduce the payoff functions of the classical Parrondo games. Call such quantum multiplexers *mundane.* In other words, mundane quantum multiplexers reside in the image of the embeddings of type 1 or 2. However, the set of quantum multiplexers is much larger than the image of embeddings of either type; that is, there are quantum multiplexers that are outside such an image. Call such quantum multiplexers *exotic.*

Clearly, the answer to question 1) above is in the affirmative for mundane quantum multiplexers based on the results of chapter 3. By taking quantum superpositions of the mundane quantum multiplexers associated with classical Parrondo games, the payoff function of the classical game can be reproduced by choosing a particular evaluative initial state such that the game is winning, even when the individual quantum games were losing with respect to appropriate evaluative initial states. For exotic quantum multiplexers, the answer is not clear cut since it is not known what an evaluative initial state for such a multiplexer should be. Therefore, a future study toward answering question 1) requires efforts into identifying such an appropriate initial state for exotic quantum multiplexers. In the context of quantum logic synthesis, how might an arbitrary quantum logic gate be synthesized via decomposition in a game theoretically meaningful way? Tha it, first assign a fixed number of qubits in the circuit to each player. Then, for an arbitrary quantum logic gate U, how might U be decomposed into sets of one qubit gates, one for each player, and an initial state choosen, such that a given game theoretic outcome might be realized?

REFERENCES

[1] A. Ahmed, S. A. Bleiler, and F. S. Khan. *Three player, Two Strategy, Maximally Entangled Quantum Games*. Proceeding of the 9$^{\text{th}}$ International Pure Math Conference, Islamabad, Pakistan, 2008. 9, 90

[2] S. D. Bartlett, H. de Guise, and B. C. Sanders. *Quantum encodings in spin systems and harmonic oscillators*. Physical Review A, Volume 65, Issue 5, 2002. 68

[3] K. Binmore. *Fun and Games: A Text on Game Theory*. D.C. Heath, 1991. 21

[4] A. Bjorck and G. H. Golub. *Numerical Methods for Computing Angles between Linear Subspaces*. Mathematics of Computation Volume 27, pages 579-594, 1973. 70

[5] S. A. Bleiler. *A Formalism for Quantum Games and an Application*. Proceeding of the 9$^{\text{th}}$ International Pure Math Conference, Islamabad, Pakistan, 2008. 2, 21

[6] S. Bone and M. Castro. *A Brief History of Quantum Computing*. Imperial College London, http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/spb3s. 3

[7] S. Bullock, D. P. O'Leary, and G. K. Brennen. *Asymptotically Optimal Quantum Circuits for d-level Systems*. Physical Review Letters, Volume 94, 230502, 2005. 69

[8] J. Daboul, X. Wang, and B. Sanders. *Quantum Gates on Hybrid Qudits*. Journal of Physics A: Mathematical and General, pages 2525-2536, 2003. 68

[9] D. Deutsch and R. Jozsa. *Rapid Solutions of Problems by Quantum Computation*. Proceedings of the Royal Society of London A, 439, pages 553558, 1992. 2

[10] J. Eisert, M. Wilkens, and M. Lewenstein. *Quantum Games and Quantum Strategies*. Physical Review Letters, Volume 83, pages 3077-3080, 1999. 2

[11] A. P. Flitney, J. Ng, and D. Abbott. *Quantum Parrondos Games*. Physica A, Volume 314, pages 35-42, 2002. 32, 33, 45, 47, 52

[12] G. H. Golub and C. F. Van Loan. *Matrix computations.* John Hopkins University Press, 1989. 79

[13] L. K. Grover. *Quantum Mechanics Helps in Searching for a Needle in a Haystack.* Physical Review Letters, Volume 79, pages 325-328, 1997. 2, 14

[14] G. P. Harmer and D. Abbott. *A Review of Parrondo's Paradox.* Fluctuation and Noise Letters, Volume 2, Number 2, 2002. 33

[15] R. J. Kay and N. F.Johnson. *Winning combinations of History-Dependent Games.* Physical Review E 67, Issue 5, 2003. 43

[16] F. S. Khan and M. A. Perkowski. *Synthesis of Ternary Quantum Logic Circuits by Decomposition.* Proceedings of the 7th International Symposium on Representations and Methodology of Future Computing Technologies, pages 114-117, 2005. 69, 71, 75

[17] F. S. Khan and M. A. Perkowski. *Synthesis of Multi-qudit Hybrid and d-Valued Quantum Logic Circuits by Decomposition.* Theoretical Computer Science, Volume 367, Issue 3, pages 336-346, 2006. 46, 69

[18] S. E. Landsburg. *Nash Equilibria in Quantum Games.* PUniversiy of Rochester, Working paper No. 524, http://www.rcer.econ.rochester.edu/RCERPAPERS/, 2006. 2, 9, 90

[19] L. Marinatto and T. Weber. *A Quantum Approach to Static Games of Complete Information.* Physical Letters A, Volume 272, Issues 5-6, pages 291-303, 2000. 2

[20] D. A. Meyer. *Quantum Strategies.* Physical Review Letters, Volume 82, pages 1052-1055, 1999. 47, 53

[21] D. A. Meyer. *Noisy Quantum Parrondo Games.* Proceedings of SPIE, Volume 5111, page 344, 2003. 33

[22] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa. *Quantum Circuits for General Multiqubit Gates.* Physical Review Letters, Volume 93, Number 13, 130502, 2004. 69, 71

[23] A. Muthukrishnan and C. R. Stroud Jr. *Multi-valued Logic Gates for Quantum Computation.* Physical Review A, Volume 62, 052309, 2000. 68, 77

[24] R. B. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, 1991. 21

[25] J. Nash. *Equilibrium Points in n-Person Games.* Proceedings of the National Academy of Sciences of the United States of America, Volume 36, Issue 1, pages 48-49, 1950. 21, 26

[26] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000. 3

[27] M. Oskin. *Quantum Computing-Lecture Notes.* Department of Computer Science and Engineering, University of Washington, http://www.cs.washington.edu/homes/oskin. 3, 19

[28] C. C. Paige and M. Wie. *History and generality of the CS decomposition.* Linear Algebra and its Applications, Volume 208-209, pages 303-326, 1994. 70

[29] J. M. R. Parrondo, G. Harmer, and D. Abbott. *New Paradoxical Games Based on Brownian Ratchets.* Physical Review Letters, Volume 85, 5226, 2000. 32, 33, 37

[30] Juan M. R. Parrondo, Gregory P. Harmer, and Derek Abbott. *New Paradoxical Games Based on Brownian Ratchets.* Physical Review Letters, Volume 85, Number 24, 2000. 44, 45

[31] V. Shende, S. Bullock, and I. Markov. *Synthesis of Quantum Logic Circuits.* IEEE Transactions on Computer Aided Design, Volume 25, Number 6, pages 1000-1010, 2006. 69, 71

[32] P. W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* SIAM Journal on Scientific and Statistical Computing, 1997. 2, 14

[33] G. W. Stewart. *Computing the CS Decomposition of a Partitioned Orthogonal Matrix.* Numerische Mathematik, Volume 40, pages 297-306, 1982. 70

[34] G. W. Stewart and J. Sun. *Matrix perturbation theory.* Academic Press Inc, 1990. 70, 112

[35] T. Tilma and E. C. G. Sudarshan. *Generalized Euler Angle Parameterization for SU(N).* Journal of Physics A: Mathematics and General, Volume 35, pages 10467-10501, 2002. 69

[36] R. Tucci. *Rudimentary Quantum Compiler.* http://arxiv.org/abs/quant-ph/9805015, 1998. 71

Appendix A

QUATERNIONS

Complex numbers are extension of real numbers. This fact motivates us to view quaternions as extension of the complex numbers, with the exception that the recipe for constructing the conjugate of a complex number needs modification when one tries to follow it to construct the conjugate of a quaternion. This modification is such that the quaternionic product is necessarily non-commutative and satisfies $zj = j\bar{z}$ for any complex number z and the quaternion j.

A.1 Complex Numbers

The set of *complex numbers* is

$$\mathbb{C} = \left\{a_0 + a_1 x : a_0, a_1 \in \mathbb{R} \text{ and } x^2 = -1\right\}.$$

Since complex numbers are just first degree polynomials, one defines binary operations of addition and multiplication on $\mathbb{C}$ via polynomial addition and multiplication respectively.

$$\begin{aligned} \text{Addition}: \quad & (a_0 + a_1 x) + (b_0 + b_1 x) = (a_0 + b_0) + (a_1 + b_1)x \\ \text{Multiplication}: \quad & (a_0 + a_1 x)(b_0 + b_1 x) = a_0 b_0 + a_1 b_0 x + a_0 b_1 x + a_1 b_1 x^2 \end{aligned}$$

The constraint $x^2 = -1$ provides multiplicative closure to $\mathbb{C}$, yielding

$$(a_0 + a_1 x)(b_0 + b_1 x) = (a_0 b_0 - a_1 b_1) + (a_0 b_1 + a_1 b_0)x$$

The equation $x^2 = -1$ has exactly two solutions, $x = \sqrt{-1}$ and $x = -\sqrt{-1}$. Setting $x = i = \sqrt{-1}$ leads to the conventional notation for the complex numbers

$$\mathbb{C} = \left\{ a_0 + a_1 i : a_0, a_1 \in \mathbb{R} \text{ and } i^2 = -1 \right\}.$$

The solutions i and $-i$ are called *imaginary numbers*. This terminology gives rise to the notion of the *real* part a_0 and the *imaginary* part a_1 of the complex number $a_0 + a_1 i$. Note that since $-i$ is also a solution to the equation $x^2 = -1$, there are complex numbers in $\mathbb{C}$ of the form

$$a_0 + a_1(-i) = a_0 - a_1 i.$$

The latter is called the *conjugate* of the complex number $a_0 + a_1 i$, and one checks that

$$(a_0 + a_1 i)(a_0 - a_1 i) = a_0^2 + a_1^2 \in \mathbb{R}$$

Clearly, the conjugate of $a_0 - a_1 i$ is the complex number $a_0 + a_1 i$; that is, double conjugation gives back the original complex number. The quantity

$$|a_0 + a_1 i| = \sqrt{a_0^2 + a_1^2}$$

defines the *length* of the complex number $a_0 + a_1 i$ (and of $a_0 - a_1 i$). It is an easy exercise to show that $\mathbb{C}$ in fact forms a field.

A.2 Quaternions

The set of *quaternions* is

$$\mathbb{H} = \left\{ p_0 + p_1 y : p_0, p_1 \in \mathbb{C} \text{ and } y^2 = -1 \right\}.$$

Again, addition and multiplication in $\mathbb{H}$ is defined as polynomial addition and multiplication, giving

$$\begin{aligned} \text{Addition}: &\quad (p_0 + p_1 y) + (q_0 + q_1 y) = (p_0 + q_0) + (p_1 + q_1) y \\ \text{Multiplication}: &\quad (p_0 + p_1 y)(q_0 + q_1 y) = p_0 q_0 + (p_1 q_0 + p_0 q_1) y + p_1 q_1 y^2 \end{aligned}$$

Is $\mathbb{H}$ closed under multiplication? The answer is yes once we note that p_0, p_1, q_0, q_1 are all complex numbers and that this requires the use of both the constraints $y^2 = -1$ and $x^2 = -1$ in simplifying the quaternionic product. Let

$$p_0 = p_0' + p_1' i, \quad p_1 = p_0'' + p_1'' i, \quad q_0 = q_0' + q_1' i, \quad q_1 = q_0'' + q_1'' i$$

be complex numbers. Simplifying the quaternionic product now results in the expression

$$
\begin{aligned}
&(p_0 + p_1 y)(q_0 + q_1 y) \\
&= (p_0'q_0' - p_1'q_1') + (p_0'q_1' + p_1'q_0')i \\
&+ \left[(p_0''q_0' - p_1''q_1') + (p_0''q_1' + p_1''q_0')i + (p_0'q_0'' - p_1'q_1'') + (p_0'q_1'' + p_1'q_0'')i\right] y \\
&+ \left[(p_0''q_0'' - p_1''q_1'') + (p_0''q_1'' + p_1''q_0'')i\right] y^2 \\
&= (p_0'q_0' - p_1'q_1' - p_0''q_0'' - p_1''q_1'') + (p_0'q_1' + p_1'q_0' - p_0''q_1'' - p_1''q_0'')i \\
&+ (p_0''q_0' - p_1''q_1' + p_0'q_0'' - p_1'q_1'')y + (p_0''q_1' + p_1''q_0' + p_0'q_1'' + p_1'q_0'')iy \\
&= (p_0'q_0' - p_1'q_1' - p_0''q_0'' - p_1''q_1'') + (p_0'q_1' + p_1'q_0' - p_0''q_1'' - p_1''q_0'')i \\
&+ \left[(p_0''q_0' - p_1''q_1' + p_0'q_0'' - p_1'q_1'') + (p_0''q_1' + p_1''q_0' + p_0'q_1'' + p_1'q_0'')i\right] y \\
&= z_0 + z_1 y
\end{aligned}
$$

for complex numbers

$$z_0 = (p_0'q_0' - p_1'q_1' - p_0''q_0'' - p_1''q_1'') + (p_0'q_1' + p_1'q_0' - p_0''q_1'' - p_1''q_0'')i$$

and

$$z_1 = (p_0''q_0' - p_1''q_1' + p_0'q_0'' - p_1'q_1'') + (p_0''q_1' + p_1''q_0' + p_0'q_1'' + p_1'q_0'')i.$$

It is important to note here that even though the variable y is a square root of -1, it is *not* equal to $\pm i$. For if it were equal to $\pm i$, then the set $\mathbb{H}$ would equal the set $\mathbb{C}$! By analogy with the complex numbers, the variable y might appropriately be called an *imaginary complex number*. It is commonly known as a *hypercomplex number*. Following convention, we replace y with j and write quaternions as $p_0 + p_1 j$.

We next develop the notion of a conjugate of a quaternion; that is, for a given quaternion p, find a quaternion q such that $pq \in \mathbb{R}$. Following the recipe that led to the definition of the complex conjugate naively we set $p_0 + p_1(-j) = p_0 - p_1 j$ as the *quaternionic conjugate* of the quaternion $p_0 + p_1 j$. This gives

$$(p_0 + p_1 j)(p_0 - p_1 j) = p_0^2 + p_1 p_0 j - p_0 p_1 j + p_1^2. \tag{A.1}$$

Multiplication of a complex number by its conjugate results in a real number that is the sum of the squares of two real numbers, namely the real and imaginary parts of the complex number. Since our definition of the quaternionic conjugate is motivated by the complex conjugate, we expect the right hand side of equation (A.1) to equal to the real number that results from the squares of the complex numbers p_0 and p_1. However, the fact that in general the square of a complex number is another complex number puts a kink in our plans. But all is not lost. Instead of insisting on the squares of the complex numbers p_0 and p_1 in our definition of the quaternionic conjugate, we are perfectly happy to work with the *squares of the lengths* of the complex numbers p_0 and p_1, which are both real numbers. This flexibility forces us to modify the proposed quaternionic conjugate to the quaternion $(\overline{p_0} - \overline{p_1} j)$ which gives

$$(p_0 + p_1 j)(\overline{p_0} - \overline{p_1} j) = |p_0|^2 + p_1 \overline{p_0} j - p_0 \overline{p_1} j + |p_1|^2 \tag{A.2}$$

To eliminate the quaternionic part from the right hand side of equation (A.2) we are forced to set

$$p_1\overline{p_0} = p_0\overline{p_1} = \overline{p_1}p_0$$

which means that $p_1\overline{p_0}$ is in fact a real number, sacrificing the generality of our argument.

At this stage, one wonders whether the recipe for the complex conjugate that has been followed thus far with a slight modification to develop the quaternionic conjugate needs to be changed drastically. Indeed, if we leave out the major ingredient of commutativity from the recipe and assume that for a complex number z,

$$zj = j\overline{z}, \tag{A.3}$$

then equation (A.1) must be re-written as

$$\begin{aligned}(p_0 + p_1j)(p_0 - p_1j) &= p_0^2 + p_1jp_0 - p_0p_1j + p_1jp_1j \\ &= p_0^2 + p_1\overline{p_0}j - p_0p_1j + p_1\overline{p_1}jj \\ &= p_0^2 + p_1\overline{p_0}j - p_0p_1j + |p_1|^2\end{aligned}$$

The occurrence of $|p_1|^2$ in the preceding equation is glaring, and suggests that we modify the proposed quaternionic conjugate yet again to be $\overline{p_0} - p_1j$ which upon multiplication with $p_0 + p_1 j$ and after using the non-commutativity condition $zj = j\overline{z}$ leads to

$$\begin{aligned}(p_0 + p_1 j)(\overline{p_0} - p_1 j) &= |p_0|^2 + p_1 j \overline{p_0} - p_0 p_1 j + |p_1|^2 \\ &= |p_0|^2 + p_1 p_0 j - p_0 p_1 j + |p_1|^2 \\ &= |p_0|^2 + |p_1|^2 \\ &= (p_0')^2 + (p_1')^2 + (p_0'')^2 + (p_1'')^2\end{aligned}$$

The quaternionic conjugate defined this way behaves much like the complex conjugate. For example, the quaternionic conjugate of $\overline{p_0} - p_1 j$ is $p_0 + p_1 j$. Moreover, as with the complex conjugate, the product of a quaternion with its conjugate is expressible as the sum of squares of four real numbers. We use the latter to define the *lenght* of a quaternion as

$$|p_0 + p_1 j| = \sqrt{|p_0|^2 + |p_1|^2} = \sqrt{(p_0')^2 + (p_1')^2 + (p_0'')^2 + (p_1'')^2}.$$

Rewriting $p_0 + p_1 j$ as

$$p_0 + p_1 j = (p_0' + p_1' i) + (p_0'' + p_1'' i) j = p_0' + p_1' i + p_0'' j + p_1'' ij \tag{A.4}$$

introduces the term ij which the non-commutativity condition of equation (A.3) shows to be a square root of -1. For convenience, set $k = ij$. Then one computes

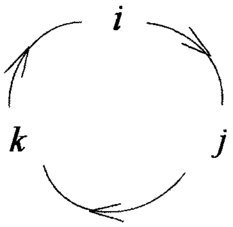
$$k^2 = (ij)^2 = (ij)(ij) = (ij)(j(-i)) = i(-1)(-i) = i^2 = -1$$

Complex number arithmetic together with equation (A.3) establish the following identities as well.

$$ik = i(ij) = i^2 j = -j$$

$$jk = j(ij) = (-i)j^2 = i$$

The last two identities and the identity $ij = k$ establish the *right-hand rule* for quaternionic multiplication which is conveniently represented in the picture below. This rule is summed up in *Hamilton's Relation* $i^2 = j^2 = k^2 = ijk = -1$.



One can verify that the quaternions satisfy all the axioms of a field except commutativity, and therefore form a division ring. Our definition of the quaternions in fact shows that the quaternions form a two dimensional algebra over the complex numbers with basis $\{1, j\}$. Equation (A.4) shows that the quaternions form a four dimensional algebra over the reals with basis $\{1, i, j, k\}$.

Appendix B

COSINE SINE DECOMPOSITION OF UNITARY MATRICES

As we shall see, the cosine sine decomposition (CSD) is essentially the well known singular value decomposition (SVD) of a unitary matrix implemented at the block matrix level. The reader is cautioned that for a given matrix, the CSD is *not* unique. The material presented in this appendix is not new. The discussion of the SVD is based on lecture notes of Professor Bin Jiang at Portland State University and the CSD discussion is based on the account given in [34] on pages 37-40.

B.1 Singular Value Decomposition

Begin with the vector and matrix 2-norms, described below.

Definition B.1. The 2-norm of a vector $x \in \mathbb{C}^n$ is the function $\|\ \|_2 : \mathbb{C}^n \to \mathbb{R}$ defined by

$$\|x\|_2 = \left(x^\dagger x\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$$

Here, $x^\dagger = (x_1^*, x_2^*, \ldots, x_n^*)^T$ and $|x_i|^2 = x_i x_i^*$ for $x_i \in \mathbb{C}$.

Definition B.2. The 2-norm of a matrix $A \in \mathbb{C}^{m \times n}$ is the function $\|\ \|_2 : \mathbb{C}^{m \times n} \to \mathbb{R}$

defined by

$$||A||_2 = \max_{||x|| \neq 0} \frac{||Ax||_2}{||x||_2} = \max_{||x||_2=1} ||Ax||_2$$

Since we will not refer to any other norms that can be defined on vectors and matrices, from now on we will use the $||\ ||$ instead of the more explicit $||\ ||_2$ to simplify notation. Also, for $A \in \mathbb{C}^{m \times n}$, denote by $A^\dagger$ the conjugate transpose of A. Recall that a matrix A is *unitary* if $AA^\dagger = A^\dagger A = I$. Equivalently, the action of a unitary matrix preserves vector norm.

Lemma B.3. Vector and matrix 2-norms are invariant under unitary transformations.

Proof. Let $U \in \mathbb{C}^{n \times n}$ be a unitary transformation, and $x \in \mathbb{C}^n$. Then

$$||Ux|| = \left((Ux)^\dagger (Ux)\right)^{\frac{1}{2}} = \left(x^\dagger U^\dagger Ux\right)^{\frac{1}{2}} = \left(x^\dagger x\right)^{\frac{1}{2}} = ||x||$$

Now let $A \in \mathbb{C}^{m \times n}$. Then

$$||AU|| = \max_{||x||=1} ||AUx|| = \max_{||Ux||=1} ||AUx|| = \max_{||y||=1} ||Ay|| = ||A||$$

If $A \in \mathbb{C}^{n \times n}$. Then

$$||UA|| = \max_{||x||=1} ||(UA)x|| = \max_{||x||=1} ||U(Ax)|| = \max_{||x||=1} ||Ax|| = ||A||$$

□

We are now ready to prove the existence of a singular value decomposition.

Proposition B.4. If $A \in \mathbb{C}^{m\times n}$, then there exists unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$, and a matrix $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p, 0, \ldots, 0) \in \mathbb{R}^{m\times n}$, $p = \min(m, n)$, such that

$$A = U\Sigma V^{\dagger}.$$

The σ_i are called *singular values* of A and are typically ordered so that

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \geq 0.$$

Proof. The proof will be inductive. Let $\sigma = \|A\|$. Since

$$\|A\| = \max_{\|x\|=1} \|Ax\|,$$

there exists a unit norm $x \in \mathbb{C}^n$ such that $\sigma = \|Ax\|$; therefore, $Ax = \sigma y$ for some $y \in \mathbb{C}^m$ with $\|y\| = 1$.

If

$$V_1 = \begin{pmatrix} v_1 & v_2 & \ldots & v_r \end{pmatrix} \in \mathbb{C}^{m\times r}, \quad r < m$$

has orthonormal columns v_i, then applying Gram-Schimdt process we can always find

$$V_2 = \begin{pmatrix} v_{r+1} & v_{r+2} & \ldots & v_m \end{pmatrix} \in \mathbb{C}^{m\times(m-r)}$$

so that (V_1, V_2) is unitary and $\text{rank}(V_1)^{\perp} = \text{rank}(V_2)$. From this fact we conclude that there exist $V_1' \in \mathbb{C}^{n\times(n-1)}$ and $V_1' \in \mathbb{C}^{m\times(m-1)}$ such that $V_1 = (x \quad V_1') \in \mathbb{C}^{n\times n}$ and

$U_1 = (y \quad U_1') \in \mathbb{C}^{m\times(m-1)}$ are unitary. Hence,

$$\begin{aligned}
U_1^{\dagger} A V_1 = \begin{pmatrix} y^T \\ U_1^T \end{pmatrix} A(x \quad V_1') &= \begin{pmatrix} y^T A x & y^T A V_1' \\ (U_1')^T A x & (U_1')^T A V_1' \end{pmatrix} \\
&= \begin{pmatrix} y^T \sigma y & y^T A V_1' \\ (U_1')^T \sigma y & (U_1')^T A V_1' \end{pmatrix} \\
&= \begin{pmatrix} \sigma & w^T \\ 0 & B \end{pmatrix} \equiv A_1
\end{aligned}$$

where $w^T \in \mathbb{R}^{(n-1)}$.

In fact $w = 0$. For by lemma **B.3.**, $||A_1|| = ||A|| = \sigma$ and

$$\begin{aligned}
||A_1|| &= \max_{||x||\neq 0} \frac{||A_1 x||}{||x||} \\
&\geq \frac{\left\| A_1 \begin{pmatrix} \sigma \\ w \end{pmatrix} \right\|}{\left\| \begin{pmatrix} \sigma \\ w \end{pmatrix} \right\|} \\
&= \frac{\left\| \begin{pmatrix} \sigma^2 + w^T w \\ Bw \end{pmatrix} \right\|}{\sqrt{\sigma^2 + w^T w}} \\
&\geq \frac{\sqrt{(\sigma^2 + w^T w)^2}}{\sqrt{\sigma^2 + w^T w}} \\
&= \sqrt{\sigma^2 + w^T w}
\end{aligned}$$

Therefore, $\sigma \geq \sqrt{\sigma^2 + w^T w}$ and hence $w^T w = 0$ which implies that $w = 0$.

We now have that

$$U_1^\dagger A V_1 = \begin{pmatrix} \sigma & 0 \\ 0 & B \end{pmatrix} \tag{B.1}$$

Now applying the same method to B and the resulting blocks B' inductively, we have

$$U_p^\dagger \ldots U_2^\dagger U_1^\dagger A V_1 V_2 \ldots V_p = \mathrm{diag}(\sigma_1, \sigma_2, \ldots \sigma_p, 0, \ldots, 0)$$

Let $U = U_1 U_2 \ldots U_p$ and $V V_1 V_2 \ldots V_p$. Then both U and V are unitary and

$$A = U \Sigma V^\dagger.$$

□

B.2 Cosine Sine Decomposition

Proposition B.5. Let the unitary matrix $W \in \mathbf{C}^{n \times n}$ be partitioned in 2×2 block form as

$$W = \begin{array}{cc} & \begin{array}{cc} l & n-r \end{array} \\ \begin{array}{c} l \\ n-l \end{array} & \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \end{array}$$

with $2l \leq n$. Then there exist unitary matrices $U = \mathrm{diag}(U_{11}, U_{22})$ and $V = \mathrm{diag}(V_{11}, V_{22})$

with $U_{11}, V_{11} \in \mathbb{C}^{l \times l}$ such that

$$U^{\dagger}WV = \begin{array}{c} \\ l \\ l \\ n-2l \end{array} \begin{array}{c} \begin{array}{ccc} l & l & n-2l \end{array} \\ \begin{pmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & I \end{pmatrix} \end{array} \tag{B.2}$$

where

$$C = \mathrm{diag}(\cos\theta_1, \cos\theta_2, \dots, \cos\theta_l)$$

$$S = \mathrm{diag}(\sin\theta_1, \sin\theta_2, \dots, \sin\theta_l)$$

such that $\sin^2\theta_i + \cos^2\theta_i = 1$ for some θ_i, $1 \le i \le l$.

Proof. Let

$$U_{11}^{\dagger}W_{11}V_{11} = C$$

be a singular value decomposition of the block W_{11} of W and suppose that

$$C = \mathrm{diag}(C_1, I_{l-k})$$

where the diagonal elements of C_1 satisfy

$$0 \le c_1 \le c_2 \le \dots c_k < 1.$$

Note that since W is unitary, the singular values cannot be greater than 1. Clearly, the columns of the matrix

$$\begin{pmatrix} W_{11} \\ W_{21} \end{pmatrix} V_{11}$$

are orthonormal. Therefore,

$$I = \left[\begin{pmatrix} W_{11} \\ W_{21} \end{pmatrix} V_{11}\right]^{\dagger} \left[\begin{pmatrix} W_{11} \\ W_{21} \end{pmatrix} V_{11}\right] = C^2 + (W_{21}V_{11})^{\dagger}\,(W_{21}V_{11})\,;$$

that is,

$$(W_{21}V_{11})^{\dagger}(W_{21}V_{11}) = \mathrm{diag}(I - C_1^2, 0_{l-k})$$

The columns of $W_{21}V_{11}$ are orthogonal with the last $(l-k)$ of them being 0. Thus, there exists a unitary matrix $\widehat{U}_{22} \in \mathbb{C}^{(n-l)\times(n-l)}$ such that

$$\widehat{U}_{22}^{\dagger} W_{21} V_{11} = \begin{pmatrix} S \\ 0 \end{pmatrix}$$

where

$$S = \mathrm{diag}(s_1, s_2, \ldots, s_k, 0, \ldots, 0) = \mathrm{diag}(S', 0) \tag{B.3}$$

with S' consisting of k rows and the all 0's block consisting of $(r-k)$ rows. Since

$$\mathrm{diag}(U_{11}, \widehat{U}_{22})^{\dagger} \begin{pmatrix} W_{11} \\ W_{21} \end{pmatrix} V_{11} = \begin{pmatrix} C \\ S \\ 0 \end{pmatrix}$$

has orthogonal columns, it follows that for $1 \leq i \leq l$

$$c_i^2 + s_i^2 = 1. \tag{B.4}$$

In particular, S' is non-singular.

Similarly, we may determine a unitary matrix $V_{22} \in \mathbb{C}^{(n-l)\times(n-l)}$ such that

$$U_{11}^{\dagger} W_{12} V_{22} = (T, 0)$$

where $T = \mathrm{diag}(t_1, t_2, \ldots, t_l)$ with $t_i \leq 0$. Since

$$U_{11}^{\dagger}(W_{11} \quad W_{12})\mathrm{diag}(V_{11}, V_{22}) = (C \quad T \quad 0)$$

has orthogonal rows, it must be that $c_i^2 + t_i^2 = 1$, and it follows from (B.3) and (B.4)that $T = -S$.

Now set $\widehat{U} = \mathrm{diag}(U_{11}, \widehat{U}_{22})$ and $V = \mathrm{diag}(V_{11}, V_{22})$. Then it follows from the preceding discussion that

$$X = \widehat{U}^{\dagger} W V$$

can be partitioned as

$$X = \begin{array}{c|ccccc} & k & l-k & k & l-k & n-2l \\ \hline k & C_1 & 0 & -S_1 & 0 & 0 \\ l-k & 0 & I & 0 & 0 & 0 \\ k & S_1 & 0 & X_{33} & X_{34} & X_{35} \\ l-k & 0 & 0 & X_{43} & X_{44} & X_{45} \\ n-2l & 0 & 0 & X_{53} & X_{54} & X_{55} \end{array} \tag{B.5}$$

Since X is unitary and Σ_1 has positive diagonal elements, we have $X_{33} = C_1$. Moreover, X_{34}, X_{35}, X_{43}, and X_{53} are zero. Therefore, the partition of X in (B.5) can now be updated to

$$X = \begin{array}{c|ccccc} & k & l-k & k & l-k & n-2l \\ \hline k & C' & 0 & -S' & 0 & 0 \\ l-k & 0 & I & 0 & 0 & 0 \\ k & S' & 0 & C' & 0 & 0 \\ l-k & 0 & 0 & 0 & X_{44} & X_{45} \\ n-2l & 0 & 0 & 0 & X_{54} & X_{55} \end{array} \tag{B.6}$$

and the the matrix

$$U_{33} = \begin{pmatrix} X_{44} & X_{45} \\ X_{54} & X_{55} \end{pmatrix} \in \mathbb{C}^{(n-l-k)\times(n-l-k)}$$

is unitary.

Now we have

$$\mathrm{diag}(I^{(l+k)}, U_{33}^{\dagger})X = \begin{pmatrix} C_1 & 0 & -S_1 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ S_1 & 0 & C_1 & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

$$= \begin{array}{c c} & \begin{array}{ccc} l & l & n-2l \end{array} \\ \begin{array}{c} l \\ l \\ n-2l \end{array} & \begin{pmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & I \end{pmatrix} \end{array}$$

Note that

$$\mathrm{diag}(I^{(l+k)}, U_{33}^{\dagger})X = \mathrm{diag}(I^{(l+k)}, U_{33}^{\dagger})U^{\dagger}WV.$$

Hence, if we set

$$\begin{aligned} U &= \widehat{U}\mathrm{diag}(I^{(l+k)}, U_{33}) \\ &= \mathrm{diag}(U_{11}, \widehat{U}_{22})\mathrm{diag}(I^{(l)}, \mathrm{diag}(I^{(k)}, U_{33})) \\ &= \mathrm{diag}(U_{11}, \widehat{U}_{22} \cdot \mathrm{diag}(I^{(k)}, U_{33})) \\ &= \mathrm{diag}(U_{11}, U_{22}) \end{aligned}$$

Set

$$U_2 = \mathrm{diag}(I_k, \widehat{U}_3)\widehat{U}_2$$

and

$$U = \mathrm{diag}(U_1, U_2)$$

Then

$$U^{\dagger}WV = \mathrm{diag}(I_{r+k}, \widehat{U}_3)X,$$

then $U^{\dagger}WV$ has the form (4.2), where U and V are block diagonal unitary matrices. □

Appendix C

LIST OF NOTATIONS AND NOMENCLATURE

- The state space of one qubit is the two dimensional complex projective Hilbert space $\mathbb{C}P^1$. As is the convention in quantum mechanics, an element ψ of the state space is denoted in Dirac notation by $|\psi\rangle$ and is called a "ket" vector.

- $|0\rangle = (1, 0)^T$ and $|1\rangle = (0, 1)^T$ are elements of the orthonormal computational basis of $\mathbb{C}P^1$. We point out that every ket is a column vector, however, as is the case here, it is sometimes written as the transpose of the appropriate row vector for notational convinience.

- $|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle = (\psi_0, \psi_1)^T$ is a quantum superposition of the elements of the computational basis, with $|\psi_0|^2 + |\psi_1|^2 = 1$. In the language of linear algebra, $|\psi\rangle$ is a linear combination of the elements of the computational basis.

- The dual of $|\psi\rangle$ is the "bra" vector $\langle\psi| = (\overline{\psi_0} \quad \overline{\psi_1})$, where $\overline{\psi_i}$ is the complex conjugate of the complex number ψ_i. Note that a bra vector is a row vector.

- For $|\psi\rangle = (\psi_0, \psi_1)^T$ and $|\phi\rangle = (\phi_0, \phi_1)^T$ in $\mathbb{C}P^1$, their inner product is given by $(|\psi\rangle, |\phi\rangle) = (\overline{\psi_0} \quad \overline{\psi_1})(\phi_0, \phi_1)^T$ and is denoted in the bra-ket notation by $\langle\psi| \, |\phi\rangle$ or just $\langle\psi|\phi\rangle$.

- The outer product of $|\psi\rangle$ and $|\phi\rangle$ is denoted by $|\psi\rangle \langle\phi|$ and is used to construct measurment operators.
- If M is a matrix, then $M^\dagger$ is the conjugate transpose of M. If M is unitary, then $M^\dagger = M^{-1}$.
- The trace $\mathrm{trace}(A)$ of a square matrix A is the sum of its diagonal elements.