Transferring Entanglement Between Different Dimensions

Timothy Forrer
Level 4 Project, MSci Natural Sciences
Supervisor: Dr V. Kendon
Department of Physics, Durham University

Submitted: April 13, 2021

Abstract

Quantum random walks have been an area of significant interest over the past two decades within quantum computation and information. They exhibit many desirable qualities that aid quantum computations and are extremely versatile for use in the design of quantum algorithms. In this report I analyse a recent publication that uses quantum walk dynamics in order to generate higher dimensional entangled states.

Contents

1 Introduction

Quantum computing is an area of intense active research, with its sensational results often making it into mainstream media. Loosely speaking, it is a field that seeks to examine how quantum phenomena can be exploited to allow for greater and more powerful computations than currently possible using classical computers. Many schemes for quantum computation utilise qubits, the quantum analogue of the bit which is the unit of information in classical computing. The main difference between the bit and the qubit, is that the bit can only exist in one of two states at any one time, whereas the qubit, due to it's quantum nature, can exist in superposition of both the possible states. Other models utilising qudits, which exist in superposition of d states rather than just two, have been proposed as they can unlock further advantages at the cost of being more complex to implement physically.

In addition to superposition, another well known quantum phenomena that is often taken advantage of in quantum computing is *entanglement*, correlations present in quantum systems that are far stronger than possible to find in classical systems. There are a variety of protocols that require the presence of entangled qubits in order to achieve results not possible with classical computers, for example superdense coding [1], quantum key distribution [2] and quantum teleportation [3]. Higher dimensional entanglement, entanglement between qudits, further enhances the power of quantum algorithms, for example superdense coding. As such the ability to possess and manipulate entangled states in higher dimensions has further benefits but again comes with its own challenges.

Quantum walks (QWs) are powerful tools in the landscape of quantum computing. Much like their classical analogues, they exhibit many properties that are desirable for computations and are an extremely useful building block for many algorithms designed for quantum computers [4]. Specific research interest into the quantum variant stems from their very significant divergences from the classical, including different spreading speeds and ability to traverse multiple paths at once. Their power is such that QWs can simulate any quantum computation and therefore are a model for universal quantum computing [5]. There is also evidence of robust performance even when the quantum computer is not perfectly isolated from its environment, and in certain situations it has been shown that decoherences due to interactions with the environment is beneficial for a given computation [6]. Quantum walks are divided into two categories, *discrete* and *continuous* time, these labels describing the nature of the evolution of the walker as the quantum walk progresses. Continuous time QWs have been shown to solve a wide range of problems in a number of different settings, in some cases exponentially faster than a classical computer is able to [7]. However, the focus of this report will be on entangled state generation and transfer which requires more than one subspace in our system, lending itself much more readily to discrete time QWs.

A potential solution to the demanding task of generating higher dimensional entanglement has been proposed [8] which uses the dynamics of QWs to transfer lower dimensional entanglement between qubits, which is far simpler to generate, into the high dimensional qudits. Whilst this scheme can be used to some moderate degree of success, I will present an alternative devised to operate in a similar setting but utilising ancilla-based quantum computing

(AQC) to instead transfer entanglement optimally.

In this report, a primer on entanglement, and two models of quantum computing, quantum walks and ancilla-based quantum computing, is given in section 2 Following this, section 3 will focus on the protocol that uses QW dynamics to facilitate the transfer of entanglement, in particular analysing its efficiency in achieving the aim of entanglement transfer. The AQC scheme for entanglement transfer is presented in section 4. Further uses of the AQC scheme beyond the transfer of entanglement are also presented in this section. Finally, a discussion on the results presented in the report, further work and a brief summary is presented in section 5.

MOVE TO WHEN NEEDED As mentioned above, discrete time quantum walks are much more suited for the purposes of this report, therefore future references to quantum walks will be assumed to be the discrete variant unless stated otherwise. We will also use QW to denote (discrete) quantum walk. The mathematical notation used in this report follows standard conventions, in particular it should be made clear the equivalency between $|u_1, u_2\rangle \equiv |u_1\rangle |u_2\rangle \equiv |u_1\rangle \otimes |u_2\rangle$, where all forms will be used interchangeably. MOVE TO WHEN NEEDED

2 Background

2.1 Quantum Computation

2.1.1 Qubits

When describing a qubit, we say that it lives in a Hilbert space $\mathcal{H}$ spanned by two states, labelled $|0\rangle$ and $|1\rangle$ in the *computational basis*. Naturally we can also use other bases for describing our qubit state. A common alternative is the basis $\{|+\rangle, |-\rangle\}$. This basis is best known as the *Hadamard basis*, as $|+\rangle$ and $|-\rangle$ are equal to $H|0\rangle$ and $H|1\rangle$, where H is the Hadamard transform given by

$$|+\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{1}$$

$$|-\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{2}$$

2.1.2 Qudits

In quantum computing, or indeed classical computing, there is no physical constraint that necessitates the use of two level qubits or bits. In classical computing, the bit can be generalised to a unit that takes on one of d states, known as the *dit*. Similarly, the quantum dit is the *qudit*, which can be in superposition of up to d states. Many aspects of quantum computing specific to qubits generalise nicely to qudits. The computational basis is now extended from $\{|0\rangle, |1\rangle\}$ to $\{|i\rangle\}_{i=0}^{d-1}$. To distinguish kets representing states of different dimension, the notation $|\cdot\rangle_d$ will be used to denote a state of dimensions d. For compactness, operators will not have their dimension explicitly labelled, since their dimension will match the states that

they are acting on. Similar to qubits, there are many different bases that can be used to describe our qudit states. The d dimensional analogue to the Hadamard transform is the *quantum Fourier transform* is given by

$$F\left|x\right\rangle_d = \frac{1}{2^{\frac{d}{2}}}\sum_{y=0}^{d-1}\omega^{xy}\left|y\right\rangle_d, \tag{3}$$

where $\omega = e^{i\frac{2\pi}{d}}$, i.e. it is the d^{th} root of unity. Therefore the Fourier basis can be generalised to the set of states $\{|+_i\rangle\}_{i=0}^{d-1}$, where

$$\left|+_i\right\rangle_d = F\left|i\right\rangle_d. \tag{4}$$

Note that in the case $d = 2$, $F = H$, hence the Hadamard transform is actually a quantum Fourier transform.

Qudits considered in isolation are not overly useful for quantum computations and in general a collection of qudits is needed for a given algorithm. The state of a collection of n qudits, which are not neccessarily all of the same dimension, is found by taking the tensor product, denoted by $\otimes$, of states describing each of the individual qudits

$$\left|\Psi\right\rangle = \left|\psi_0\right\rangle_{d_0} \otimes \left|\psi_1\right\rangle_{d_1} \otimes \cdots \otimes \left|\psi_{n-1}\right\rangle_{d_{n-1}}. \tag{5}$$

This notation can be further simplified by the omission of the tensor product symbol,

$$\left|\Psi\right\rangle = \left|\psi_0\right\rangle_{d_0}\left|\psi_1\right\rangle_{d_1}\ldots\left|\psi_{n-1}\right\rangle_{d_{n-1}}. \tag{6}$$

Where there are qudits of equal dimension it is possible to further reduce the notation by combining kets together,

$$\begin{aligned}
&\left|\psi_0\right\rangle_d \otimes \left|\psi_1\right\rangle_d \otimes \cdots \otimes \left|\psi_{n-1}\right\rangle_d && (7)\\
=&\left|\psi_0\right\rangle_d \left|\psi_1\right\rangle_d \ldots \left|\psi_{n-1}\right\rangle_d && (8)\\
=&\left|\psi_0\psi_1\ldots\psi_{n-1}\right\rangle_d. && (9)
\end{aligned}$$

Since tensor products distribute over sums, when combining kets in superpositions they must also be distributed over. For example with two qudits,

$$\begin{aligned}
&(\left|0\right\rangle_d + \left|1\right\rangle_d) \otimes \left|2\right\rangle_d && (10)\\
=&(\left|0\right\rangle_d + \left|1\right\rangle)_d \left|2\right\rangle_d && (11)\\
=&\left|02\right\rangle_d + \left|12\right\rangle_d && (12)
\end{aligned}$$

Recall also from the properties of the tensor product that it is not commutative, therefore care must be taken in ensuring that the order of qudits is preserved throughout,

$$\left|01\right\rangle_2 \neq \left|10\right\rangle_2. \tag{13}$$

In this report all three methods of expression will be used, with clarity given due preference over compactness. The subscripts denoting the dimension of a qudit state will also be dropped when their dimension has already been explictly stated.

2.1.3 Operators

There are a wide variety of other operators beyond those introduced above that equate to a change of bases. One of the most common is the (Pauli) X, or NOT, operator. As the second name implies, in the qubit setting acts in the same way as the classical NOT operator,

$$|0\rangle \mapsto |1\rangle \tag{14}$$
$$|1\rangle \mapsto |0\rangle . \tag{15}$$

More generally in the qudit setting,

$$X |i\rangle_d = |i + 1 \bmod d\rangle_d . \tag{16}$$

However, there are other gates that do not have a classical analogue such as the the (Pauli) Z gate. In the qubit setting acts Z as the identity on $|0\rangle$ and applies a relative phase to $|1\rangle \mapsto -|1\rangle$. Again this can be generalised to the qudit setting

$$Z |i\rangle_d = \omega^i |i\rangle_d , \tag{17}$$

where ω is again the d^{th} root of unity. Both of these operators are sometimes referred to as Pauli operators as their matrix representations in the computational basis are given by the Pauli matrices σ_x and σ_z. There is also the Y operator corresponding to the Pauli σ_y matrix but this is often omitted from consideration as

$$ZX = iY. \tag{18}$$

Hence when designing a set of operators for a quantum computer, there is no theoretical need for Y as it already exists up to a global phase, which is essentially irrelevant. The final operation to be introduced in the section is the C-U, or controlled-U operator, where U is any unitary operation. This operator is distinct from the others previously introduced in that it acts on two qudits rather than just one, although it only directly alters the state of one of the two qudits. In the qubit setting, one qubit is designated as the control qubit and the other the target. If the control qubit is in the state $|0\rangle$, then nothing happens to the target qubit. If the control qubit is in the state $|1\rangle$, then the operator U is applied to the target qubit. Written mathematically, where the left hand qubit is the control, the right hand qubit is the target in some arbitrary qubit state $|\psi\rangle$,

$$C\text{-}U\left(|0\rangle \otimes |\psi\rangle\right) = |0\rangle \otimes |\psi\rangle \tag{19}$$
$$C\text{-}U\left(|1\rangle \otimes |\psi\rangle\right) = |1\rangle \otimes U |\psi\rangle . \tag{20}$$

To extend this to the qudit setting, first note that equations 19-20 can be compactly written as

$$C\text{-}U\left(|x\rangle \otimes |\psi\rangle\right) = |x\rangle \otimes U^x |\psi\rangle . \tag{21}$$

This equation now is well-defined for qudits, where instead of just taking values 0 or 1, x can now take values from $\{0, 1, \ldots, d-1\}$. Note that this definition applies also when the dimensions of the control and target are not equal, provided that the action of U on the target is well-defined

$$C\text{-}U\left(|x\rangle_d \otimes |\psi\rangle_{d'}\right) = |x\rangle_d \otimes U^x |\psi\rangle_{d'} . \tag{22}$$

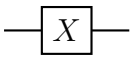
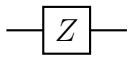
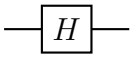
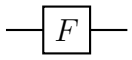
Operator	Gate
(Pauli) X	
(Pauli) Z	
Hadamard	
QFT	

Table 1: Table of operators and their gate representations in quantum circuits.

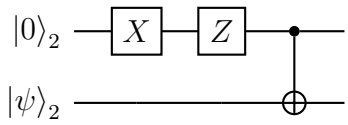


Figure 1: An example of a circuit schematic representing the expression given in equation 24.

In this report, unless otherwise stated, it is assumed that control qudits are always on the left and target qubits on the right.

It is sometimes neccessary to have operators that act on single qudits in a collection of n qudits. These can be expressed by taking the tensor product with the identity acting on the other qudits in the collection. For example, the operator that acts with an X on the qudit of index 1 in the state $|\psi_0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle$ is given by

$$I \otimes X \otimes I. \tag{23}$$

2.1.4 Circuits and Gates

When describing a circuit that implements a series of operators on our qudits, it is often much clearer and simpler to use a circuit diagram. Straight lines in the circuits represent qubits/qudits. Operations on qudits are represented by gates placed on the straight lines corresponding to the qudits we wish to act on. A table of gates relevant for this report is given in table 1.

When reading a circuit, time flows from left to right and a gate must finish its operation on a qudit before the next gate on the same qudit is enacted. Figure 1 is an example of a two qubit circuit implementing the following series of operations

$$C\text{-}X(Z \otimes I)(X \otimes I)\left(|0\rangle_2 \otimes |\psi\rangle_2\right). \tag{24}$$

2.2 Entanglement

Entanglement is a property of quantum systems that is the source of many of the remarkable results displayed by quantum algorithms. Essentially it is the presence of correlations in a

quantum system that is stronger than possible classically. Although entanglement can exist between multipartite systems, such as in the case of GHZ states or W states, in this report only bipartite entanglement will be discussed. Therefore, all references to entanglement are specifically referring to bipartite entanglement. Generalising entanglement to higher dimensions in the bipartite setting is done by having entanglement between qudits as opposed to qubits. There is no requirement for the dimensions of the entangled qudits to be matching. For example,

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle_2 \otimes |0\rangle_3 + |1\rangle_2 \otimes |1\rangle_3 \right), \tag{25}$$

is a entangled state between a qubit and a qutrit ($d = 3$).

2.2.1 Measuring entanglement

In order to analyse the efficiency of an entanglement transfer protocol, it is useful to have a method of quantifying entanglement, of which numerous methods have been proposed and are used. For a full mathematical discussion on entanglement quantification, see [9]. Here two methods of quantifying entanglement are defined.

Definition 2.1 (Negativity)**.** *Let ρ be a density matrix describing a bipartite quantum system $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$. The negativity, $\mathcal{N}$, of ρ is defined as*

$$\mathcal{N}(\rho) = \frac{||\rho^{\Gamma_A}||_1 - 1}{2}, \tag{26}$$

where ρ^{Γ_A} denotes the partial transpose of ρ with respect to the subsystem $\mathcal{H}_A$ and $\|X\|_1 = Tr(\sqrt{X^\dagger X})$ denotes the trace norm of X.

The second method is closely related to the negativity and is called the *logarithmic*, or *log*, *negativity*.

Definition 2.2 (Logarithmic Negativity)**.** *Let ρ be the same as in definition 2.1. The logarithmic negativity, $E_{\mathcal{N}}(\rho)$, of ρ is defined as*

$$E_{\mathcal{N}}(\rho) = \log_2 \left\| \rho^{\Gamma_A} \right\|_1. \tag{27}$$

ρ^{Γ_A} and $\|X\|_1$ denote the same operations as definition 2.1.

From these definitions the relationship between $\mathcal{N}$ and $E_{\mathcal{N}}$ is easy to derive,

$$E_{\mathcal{N}} = \log_2 (2\mathcal{N} + 1). \tag{28}$$

One key difference between these two measures is they way in which they quantify entanglement of maximally entangled states. Bell states, which are maxially entangled two qubit states, have a negativity of $\frac{1}{2}$, and a log negativity of 1. Consider now the maximally entangled ququart state ($d = 4$),

$$\frac{1}{2} \left(|00\rangle + |11\rangle + |22\rangle + |33\rangle \right). \tag{29}$$

This state can be considered equivalent to two Bell states (see Appendix XXX). Therefore, it would be desirable for our entanglement quantifier measure the entanglement in the ququart

state to be twice the value for the Bell state, in order to make comparisons for entanglement transferred. Using the negativity, the ququart gives a value of 1.5 (three times the negativity of a Bell state). The log negativity however gives a value of 2 (twice the log negativity of a Bell state). Hence, when quantifying entanglement transferred, in this report we will use the log negativity as it will allow for direct comparison. As with all quantities that can be measured there is a unit of bipartite entanglement.

Definition 2.3 (Ebit)**.** *The ebit is defined as the entanglement present in a maximally entangled two qubit state (a Bell state).*

Note that this definition is free from specifying a measure of entanglement, so it is not always the case that 2 ebits of entanglement is present in a state equivalent to 2 Bell states (the negativity is an example of this). As shown previously though, this is the case in the log negativity and indeed, 1 ebit is equal to 1 unit of log negativity.

Claim 2.4. *A maximally entangled state in $d \times d$ dimensions has $\log_2 d$ ebits when measured with log negativity.*

Proof. First, note that for a Hilbert space of dimension $d \times d$, the state

$$|\psi\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{j=d-1} |j\rangle \otimes |j\rangle \tag{30}$$

is maximally entangled, since its reduced density matrix is maximally mixed. Therefore, to calculate the entanglement of a maximally mixed state it is sufficient to measure the entanglement of $|\psi\rangle$. Since $|\psi\rangle$ is a pure state, its density matrix is given by

$$\rho = |\psi\rangle \langle\psi| \tag{31}$$

$$= \frac{1}{d} \sum_{j,k} |j\rangle \langle k| \otimes |j\rangle \langle k| \tag{32}$$

To calculate $E_{\mathcal{N}}$ we need the partial transpose on one of the subsystems, the choice of which is irrelevant since they are both entangled with one another. So we choose to find the partial transpose on A.

$$\rho^{\Gamma_A} = \frac{1}{d} \sum_{j,k} |k\rangle \langle j| \otimes |j\rangle \langle k| \tag{33}$$

From this we wish to calculate the trace norm

$$\|\rho\|_1 = \sqrt{\rho\rho^\dagger}. \tag{34}$$

$$\rho^{\Gamma_A}\left(\rho^{\Gamma_A}\right)^\dagger = \frac{1}{d^2}\sum_{j,k}|k\rangle\langle k|\otimes|j\rangle\langle j| \tag{35}$$

$$= \frac{1}{d^2}I_d\otimes I_d \tag{36}$$

$$= \frac{1}{d^2}I_{d^2} \tag{37}$$

$$\implies \|\rho^{\Gamma_A}\|_1 = \frac{1}{d}\operatorname{Tr}\left(I_{d^2}\right) \tag{38}$$

$$= \frac{1}{d}\cdot d^2 \tag{39}$$

$$= d \tag{40}$$

Therefore, this gives us that

$$E_{\mathcal{N}} = \log_2\left\|\rho^{\Gamma_A}\right\| \tag{41}$$

$$= \log_2 d. \tag{42}$$

□

2.3 Random Walks

2.3.1 Classical Random Walk

Before discussing quantum random walks, I will first motivate their design using the example of a classical random walk on a discrete number line. In the classical random walk, a walker (often described as being somewhat inebriated) is constrained to moving up and down a discrete number line, starting their walk at the origin. To determine whether to take a step to the left (-1) or the right (+1), the walkers flips an unbiased coin, moving to the right if the coin lands on heads and to the left if the coin lands on tails. The process is can be repeated over and over until a desired stopping point is reached, after a given number of coin flips, or until the walker has reached a specific destination, such as their house in the case of the inebriated walker. The walk can also continue on forever and in this limit, the walker will reach every point on the number line. The probability distribution describing the probability of the walker being in any given position away from the origin is given by a binomial distribution, shown in orange in Figure 2 for 100 coin flips.

2.3.2 Quantum Random Walk

With the classical random walk model in mind, we are now in a position to 'quantise' it into the *quantum random walk*. In our walker system, we can divide the overall Hilbert space of the QW, $\mathcal{H}$, into two subspaces, the coin subspace $\mathcal{H}_C$ and the position subspace of the walker $\mathcal{H}_W$.

$$\mathcal{H} = \mathcal{H}_C\otimes\mathcal{H}_W. \tag{43}$$

To aid distinguishability between coin states and position states, we write that

$$\langle\mathcal{H}_C\rangle = \{|\uparrow\rangle, |\downarrow\rangle\} \tag{44}$$

$$\langle\mathcal{H}_W\rangle = \{|k\rangle\,|k\in\mathbb{Z}\}, \tag{45}$$

where $\langle U \rangle$ denotes a set of vectors which span U. Therefore, the states $|\uparrow\rangle, |\downarrow\rangle$ take the place of heads and tails on our quantum 'coin'. Note in particular that there is no constraint placed on the dimensionality of $\mathcal{H}_W$, and therefore the walker can be taken to be a qudit, whereas $\dim \mathcal{H}_C = 2$ and the coin is taken to be a qubit. Having defined the Hilbert space within which the walk will be conducted, we can now define operators within our space that will dictate how the QW will proceed. We first define the 'coin flip' operator $C \in \mathcal{H}_C$. There is a continuum of choices for C, details of which can be found here [10]. For walks on a line, if we restrict ourselves to choosing an unbiased coin with real coefficients the Hadamard coin is the only choice of coin available. This takes on the same form as the Hadamard transform defined in Section 2.1.4,

$$C = \frac{1}{\sqrt{2}}\Big[|\uparrow\rangle\langle\uparrow| + |\uparrow\rangle\langle\downarrow| + |\downarrow\rangle\langle\uparrow| - |\downarrow\rangle\langle\downarrow| \Big] \tag{46}$$

$$= \frac{1}{\sqrt{2}}\Big[(|\uparrow\rangle + |\downarrow\rangle)\langle\uparrow| + (|\uparrow\rangle - |\downarrow\rangle)\langle\downarrow| \Big] \tag{47}$$

where $|0\rangle \to |\uparrow\rangle$ and $|1\rangle \to |\downarrow\rangle$. Equation 47 makes obvious the action of C; if the coin state is $|\uparrow\rangle$ then it becomes an equal superposition of $|\uparrow\rangle + |\downarrow\rangle$, if the coin state is in $|\downarrow\rangle$ then we get an equal superposition of $|\uparrow\rangle - |\downarrow\rangle$. These two equal superpositions are often denoted as $|+\rangle$ and $|-\rangle$ respectively.
We then define our shift operator $S \in \mathcal{H}$ which allows the position of our walker to change, dependent on the state of the coin.

$$S = \sum_k |\uparrow\rangle\langle\uparrow| \otimes |k+1\rangle\langle k| + |\downarrow\rangle\langle\downarrow| \otimes |k-1\rangle\langle k| . \tag{48}$$

Again, this representation of S makes manifest its effect on our walker. If the coin is in the $|\uparrow\rangle$, then we take a step in the +1 direction, if in the $|\downarrow\rangle$ then we take a step in the -1 direction. The probability distribution of such a walk is plotted in Fig 2, where the initial coin state is $|\downarrow\rangle$, and is compared to a classical random walk.

Whilst the above choice of S is the most common on the number line, it is also possible to define an alternative choice of shift operator,

$$\tilde{S} = \sum_k |\uparrow\rangle\langle\uparrow| \otimes |k\rangle\langle k| + |\downarrow\rangle\langle\downarrow| \otimes |k+1\rangle\langle k| . \tag{49}$$

Th subtle difference between $\tilde{S}$ and S is that $\tilde{S}$ can only move in the +1 direction of the number line and has no 'left moving' part, so to speak. This means that $\tilde{S}$ is restricted to the non-negative integers and unlike S, can occupy all $|x\rangle$ for $0 \leq x \leq T$, where T is the number of time steps in our QW.

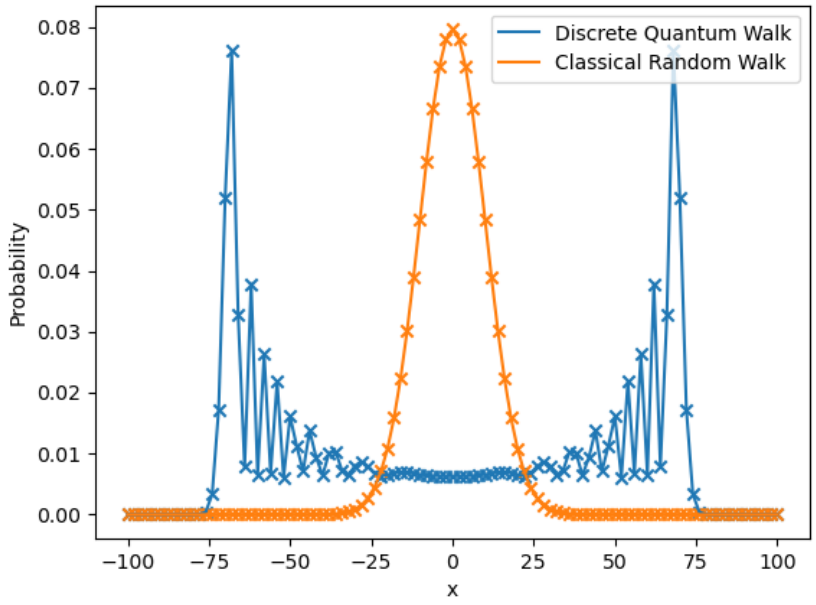


Figure 2: The probability distributions of a classical and quantum walk on a line, both originating at the origin. The quantum walk coin is initially in the $\frac{1}{\sqrt{2}}(|\uparrow\rangle + i\,|\downarrow\rangle)$ state. Odd points have been omitted as they all have zero probability.

2.4 Ancilla-Based Quantum Computing

Quantum walks is but one of many universal quantum computing models. Another model is known as *ancilla-based quantum computing*, which, in particular, aims to resolve two conflicting demands when building quantum computers: We wish for our qubits to be well isolated to prevent decoherence, yet also want them to interact with each other to perform our quantum computations. However, a qubit cannot distinguish between unwanted and wanted interactions, resulting in the need for a balancing act that adequetely addresses these two issues.

Ancilla-based quantum computing aims to resolve this conflict by using additional qubits, known as ancilla qubits, which mediate interactions between the main register qubits. Using this model, it is possible to implement two complementary registers of qubits. The main register can be designed to be strongly isolated from interactions, bar a select few natural interactions with the ancilla registry, whose qubits can be easily manipulated but decohere much faster. Quantum computations can be performed by *delocalising* quantum information from the main register across both the register and the ancilla. After performing a computation on the ancilla qubit, the quantum information is then *relocalised* back into the main register. The ancilla can be reset to the initial state and used again for other computations. A simple example of a circuit designed for ancilla-based quantum computing is shown in Figure 3.

The circuit on the left hand side of Figure 3 does not directly implement any unitary trans-

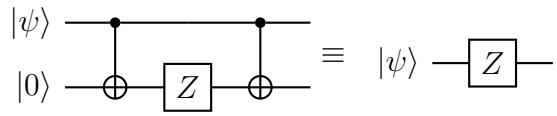


Figure 3: Two circuits that implement a Z gate acting on an arbitrary qubit $|\psi\rangle$.

formations on an arbitrary qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, but instead indirectly acts on $|\psi\rangle$ via the ancilla qubit, to which the quantum information of $|\psi\rangle$ is shared to via a CNOT gate. Acting a Z gate on the ancilla and relocalising the quantum information with another CNOT gate 'passes on' the effects of the Z gate. This model of quantum computing does not have to be reserved to qubit computing alone and applies to qudits and even quantum continous variables (QCVs). However, here the focus will be on utilising it with qubits and qudits. In fact there is no prerequisite to only using qudits of similar dimension, and defining quantum operators designed for mismatched dimensions is relatively straightforward to do. First examine the the case of the controlled NOT gate. In Section 2.1.4 it was shown that the C-X gate acts an X gate on the target qubit if the control qubit is in the state $|1\rangle$. This can be summarised as follows

$$C\text{-}X\left(|i\rangle_2 \otimes |j\rangle_2\right) = |i\rangle_2 \otimes X^i |y\rangle_2 . \tag{50}$$

Expressed in this way there is quite a natural extension of the C-X gate in the case that the control and target are both qudits of arbitrary dimension,

$$C\text{-}X\left(|i\rangle_d \otimes |j\rangle_{d'}\right) = |i\rangle_d \otimes X^i |y\rangle_{d'} , \tag{51}$$

recalling that the general X gate in d' dimensions increments the basis states by 1, mod d'. Indeed any C-U gate can be expressed in this manner

$$C\text{-}U\left(|i\rangle_d \otimes |j\rangle_{d'}\right) = |i\rangle_d \otimes U^i |y\rangle_{d'} , \tag{52}$$

provided that U has been well-defined in d' dimensions.

Later in this report I will highlight how this particular model of quantum computing is well suited for the protocol outlined in [8], which will be discussed in detail in the following section.

3 Entanglement Transfer using Quantum Walks

We now review the protocol presented in [8] which utilises quantum walk dynamics in order to generate higher dimensional entanglement.

The imagined setup of this protocol is that there are two labs, A and B, which have a shared source of entangled qubits but are otherwise spatially separated and cannot interact with one another.

In this section we use following notation:

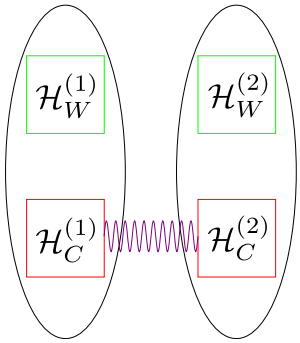


Figure 4: The initial prepared state has entanglement solely between the two coin subspaces. Figure is an edited version from FIG 3 from [8].

- $|u\rangle_J$ is a state belonging to the subspace $\mathcal{H}_J = \mathcal{H}_J^{(A)} \otimes \mathcal{H}_J^{(B)}, J \in \{C, W\}$.
- $|u\rangle^{(K)}$ is a state belonging to the subspace $\mathcal{H}^{(K)} = \mathcal{H}_C^{(K)} \otimes \mathcal{H}_W^{(K)}, K \in \{A, B\}$.
- $|u\rangle_J^{(K)}$ is a state belonging to the subspace $\mathcal{H}_J^{(K)}$.

In this mathematical framework the overall Hilbert space of the quantum system is comprised of two quantum walk subspaces,

$$\mathcal{H} = \mathcal{H}^{(A)} \otimes \mathcal{H}^{(B)} \tag{53}$$

$$= \mathcal{H}_C^{(A)} \otimes \mathcal{H}_W^{(A)} \otimes \mathcal{H}_C^{(B)} \otimes \mathcal{H}_W^{(B)}. \tag{54}$$

The basic premise of this protocol is this:

1. Entangle the two coin spaces of the walkers $\mathcal{H}_C^{(K)}$. (Fig 4.)
2. Proceed with the quantum walk for some determined number of steps.
3. Use a projection $\mathcal{P}_\gamma = |\gamma\rangle \langle\gamma|, |\gamma\rangle \in \mathcal{H}_C^{(A)}$ to then transfer the entanglement so that it solely exists in the subspace $\mathcal{H}_W^{(A)} \otimes \mathcal{H}_C^{(B)} \otimes \mathcal{H}_W^{(B)}$.
4. In similar fashion, find a projection $\mathcal{P}_\delta = |\delta\rangle \langle\delta|, |\delta\rangle \in \mathcal{H}_C^{(B)}$ to transfer the entanglement to exist between the two walker subspaces, $\mathcal{H}_W^{(i)}$, only.
5. Accumulate entanglement in the walker subspaces by once more entangling the two coin spaces and repeating the protocol.

In this way, we are able to generate arbitrary amounts of higher dimensional entanglement. As is the case with many quantum walk based protocols, particular attention must be paid to the choice of coin used for the quantum walk, as it will have a large impact on the outcomes of the protocol. The shift operator used in this protocol is the $\tilde{S}$, as outlined in Section 2.3.2.

3.1 Transfer using identity coin operator

To illustrate the basic principles of the protocol, consider the case where our coin is the identity I. A state $|\psi(0)\rangle$ is prepared with the coin states entangled and walkers at the origin

$$|\psi(0)\rangle = \underbrace{\frac{1}{\sqrt{2}}\Big[\,|\uparrow\rangle_C^{(A)}\,|\uparrow\rangle_C^{(B)} + |\downarrow\rangle_C^{(A)}\,|\downarrow\rangle_C^{(B)}\,\Big]}_{\text{Bell State}} \otimes |0\rangle_W^{(A)}\,|0\rangle_W^{(B)}\,. \tag{55}$$

Following this, the 'coin', I, is applied and then use our shift operator $\tilde{S}$ to advance the quantum walk. Explicitly (dropping the indices and combining some of our kets together) we obtain

$$|\psi(1)\rangle = \tilde{S}I\,|\psi\rangle = \frac{1}{\sqrt{2}}\Big[\,|\uparrow,\uparrow\rangle\,|0,0\rangle + |\downarrow,\downarrow\rangle\,|1,1\rangle\,\Big]. \tag{56}$$

We then project the part of $|\psi(1)\rangle$ residing in the $\mathcal{H}_C^{(A)}$ subspace onto the vector $|\gamma\rangle$, using the projective operator $\mathcal{P}_\gamma = |\gamma\rangle\langle\gamma| \in \mathcal{H}_C^{(A)}$. Choose $|\gamma\rangle = \frac{1}{\sqrt{2}}\big[\,|\uparrow\rangle + |\downarrow\rangle\,\big]$ which then gives us

$$\mathcal{P}_\gamma\,|\psi(1)\rangle = \frac{1}{2}\Big[\,|\gamma\rangle \otimes \big(\,|\uparrow,0,0\rangle + |\downarrow,1,1\rangle\,\big)\Big]. \tag{57}$$

Similarly, we project the other walker subspace to $|\delta\rangle$ which we can in this instance take to be the same state as $|\gamma\rangle = \frac{1}{\sqrt{2}}\big[\,|\uparrow\rangle + |\downarrow\rangle\,\big]$,

$$\mathcal{P}_\delta\mathcal{P}_\gamma\,|\psi(1)\rangle = \frac{1}{2\sqrt{2}}\Big[\,|\gamma\rangle \otimes |\delta\rangle \otimes \frac{1}{\sqrt{2}}\Big(\,|0,0\rangle + |1,1\rangle\,\Big)\Big]. \tag{58}$$

Renormalising we see that we have the state

$$|\gamma\rangle_C^{(A)} \otimes |\delta\rangle_C^{(B)} \otimes \underbrace{\frac{1}{\sqrt{2}}\Big[\,|0,0\rangle + |1,1\rangle\,\Big]_W}_{\text{Bell State}}, \tag{59}$$

which has a Bell State in the $\mathcal{H}_W$ subspace, and the states in $\mathcal{H}_C$ are separable. Therefore we have transferred the entanglement that originally resided in the coin subspace to the walker one.

3.2 Accumulation

The true motivation behind this protocol, as previously stated, is the ability to accumulate the entanglement transferred from the lower dimensional coin subspace to the higher dimensional walker one. Again, using I as our coin, start with the final state obtained from the first iteration of the protocol (Equation 59) and re-entangle the coin subspaces, obtaining a new initial state, $|\psi(0)\rangle$

$$|\gamma\rangle_C^{(A)} \otimes |\delta\rangle_C^{(B)} \otimes \frac{1}{\sqrt{2}}\Big[\,|0,0\rangle + |1,1\rangle\,\Big]_W \tag{60}$$

$$\xrightarrow{\text{Entangle } \mathcal{H}_C} \frac{1}{\sqrt{2}}\Big[\,|\uparrow,\uparrow\rangle + |\downarrow,\downarrow\rangle\,\Big]_C \otimes \frac{1}{\sqrt{2}}\Big[\,|0,0\rangle + |1,1\rangle\,\Big]_W \tag{61}$$

$$:= |\psi(0)\rangle\,. \tag{62}$$

We then proceed with the walk, however taking two steps instead of one this time,

$$|\psi(2)\rangle = (\tilde{S}I)^2 |\psi(0)\rangle \tag{63}$$

$$= \frac{1}{2}\Big[|\uparrow,\uparrow\rangle \big(|0,0\rangle + |1,1\rangle \big) + |\downarrow,\downarrow\rangle \big(|2,2\rangle + |3,3\rangle \big)\Big]. \tag{64}$$

Using the same projection operators in the two coin subspaces, $\mathcal{P}_\gamma \in \mathcal{H}_C^{(A)}, \mathcal{P}_\delta \in \mathcal{H}_C^{(B)}$, and renormalising gives the final state

$$|\gamma\rangle \otimes |\delta\rangle \otimes \frac{1}{2}\Big[|0,0\rangle + |1,1\rangle + |2,2\rangle + |3,3\rangle \Big]. \tag{65}$$

If we measure the entanglement in the state

$$\frac{1}{2}\Big[|0,0\rangle + |1,1\rangle + |2,2\rangle + |3,3\rangle \Big] \tag{66}$$

using the log negativity, we find that we have now accumulated two ebits in the walker subspace as desired. We can continue to repeat this process to accumulate arbitrarily large amounts of entanglement into our walker subspace. The number of steps needed for each iteration of the quantum walk is given as follows.

Claim 3.1. *The n^{th} iteration (counting from 1) of the protocol requires at least 2^{n-1} steps in the quantum walk.*

Proof. Each step in a quantum walk with shift operator $\tilde{S}$ increases the number of basis states with non-zero amplitude by 1, provided that the amplitude $|\downarrow\rangle$ coin basis state is non-zero. Therefore the dimension of each walker can be taken to be $s+1$, where s is the total number of steps taken in the walk, since the walker space can be reduced to the subspace spanned by the $s+1$ non-zero amplitude basis states. Using Claim 2.4, the upper bound on the entanglement that can be held between the two walker states is given by $\log_2(s+1)$ ebits. This implies the following condition on the total number of steps

$$\log_2(s+1) \geq n \implies s \geq 2^n - 1. \tag{67}$$

Assuming that this inequality is satisfied at equality for the $n-1^{th}$ iteration of the protocol gives a total step number of $2^{n-1}-1$. Therefore, the minimum number of steps for the n^{th} iteration is given by

$$2^n - 1 - (2^{n-1} - 1) = 2^n - 2^{n-1} \tag{68}$$

$$= 2 \times 2^{n-1} - 2^{n-1} \tag{69}$$

$$= 2^{n-1}. \tag{70}$$

□

3.3 Retrieval

Although accumulating the entanglement in higher dimensions is of significant use, it is also possible to imagine that we might wish to retrieve the entangled Bell pairs back from our qudits. For example, consider the case where the method of generating entangled qubits is not deterministic but we have a protocol which requires a large number of Bell pairs to be used at the same time. If this protocol can be reversed to get Bell pairs back out, then we could store the entangled Bell pairs when they are successfully generated and then retrieve them all at once to ensure we have an adequate number of Bell pairs. In the case of the identity coin example, it is rather simple to use a near identical setup, where the qudits play the role of the coins and the qubit play the role of the walker, to retrive entangled Bell pairs out of the qudits. For a general coin, it is also possible in principle to retrieve the entanglement back out of the entangled qudits. However, retrieval after more than 1 ebit has been transferred for the non-identity coin is not simple within the framework of a quantum walk based protocol, due to the non maximal nature of the entanglement present in the qudits.

3.4 Drawbacks

In the example given above, I have demonstrated that this protocol can transfer all of the entanglement to our qudits. However, this is with the significant caveat that the example does not use quantum walk dynamics in order to transfer the entanglement, as using the identity as a coin is akin to having no coin at all. In analysing the protocol with the Hadamard coin, it is only possible to transfer one ebit of entanglement optimally, after the first iteration. Subsequent iterations result in loss of entanglement. Simulations of the protocol for storing two Bell states, equivalent to two ebits, found that at most around 1.6 ebits of entanglement can be stored with a Hadamard QW, as shown in Figure 5. Furthermore, the projective measurements employed as part of the protocol mean that it is not unitary and is non trivial to reverse in order to retrieve the entanglement out from the entangled qudits. The experimental implementation suggested in the paper also required the use of post selection, where undesirable states were discarded and the protocol run again. All this in combination results in a protocol which is rather inefficient in achieving its aims, and serves more as a proof of concept in that it is possible to use quantum walk dynamics to transfer entanglement, but falls short in being a suitable implementation for the task.

However, the example with the identity coin clearly shows that it is possible to efficiently transfer entanglement into qudits, but highlights that an alternative computational model is likely better suited in implementing the ideas proposed. In Section 4, an alternative protocol in the ancilla-based quantum computing model is proposed which solves the inefficiencies present in this QW based proposal.

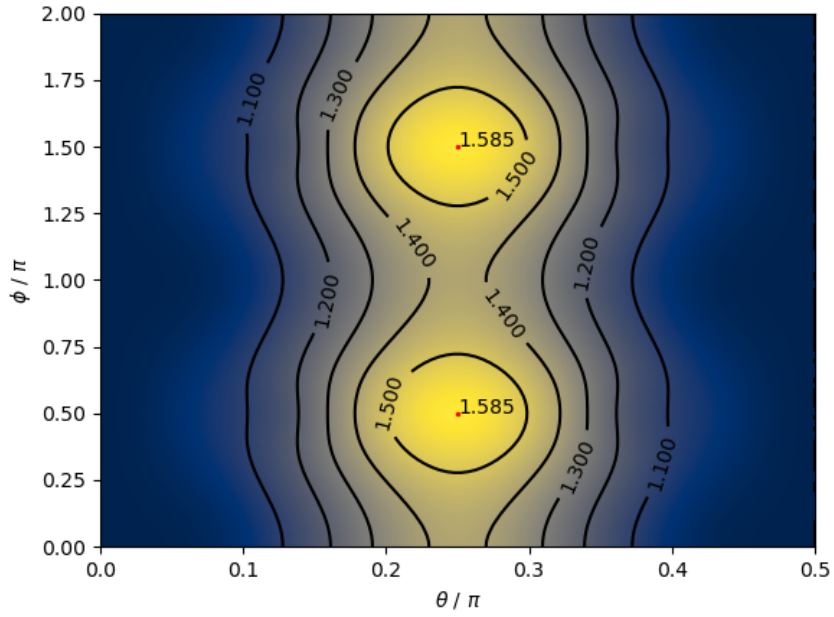


Figure 5: Transferring two ebits using the quantum walk protocol proposed by [8]. θ, ϕ are the parameters describing $|\gamma\rangle$ that defines the projection operator $\mathcal{P}_\gamma$.

4 Entanglement Transfer using Ancilla-Based Quantum Computing

In this section an alternative entanglement transfer scheme is outlined. It is designed to operate in the same setting at the QW based protocol, but instead is able to transfer entanglement deterministically and also retrieve it perfectly as well.

The protocol can be succintly summarised by the circuit schematic given in figure 6, and essentially is two copies of the same circuit.

Since the two halves of the circuit are completely separate from each other, this circuit is suitable for the same experimental setup described for the QW protocol, where it is imagined that there are two labs which are spatially separated, and have a shared source of Bell state entangled qubits which are sent individually to each lab. The majority of the gates shown in the schematic are discussed in Section 2.1.4, with the exception of the gate Ω^j. The action of Ω on the computational basis states is given by the matrix

$$\Omega = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix} \tag{71}$$

$$\implies \Omega^j = \begin{pmatrix} 1 & 0 \\ 0 & \omega^j \end{pmatrix}. \tag{72}$$

Essentially it is the similar to the Z operator in that it applies relative phase difference, but instead of a phase of -1, a phase corresponding to the d^{th} root of unity is introduced on

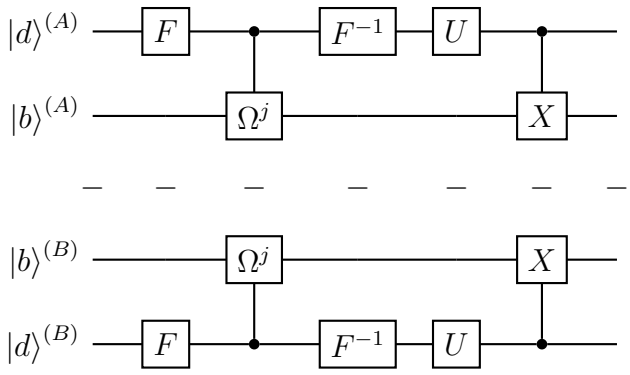


Figure 6: The AQC circuit for entanglement transfer. The schematic shows two separated circuits that exist in spatially separated labs, A and B. The qudits and qubits belonging to each individual lab are labelled by their superscripts.

the $|1\rangle_2$ states, with d corresponding to the dimension of the control qudits. Therefore C-Ω operator is one that acts as

$$C\text{-}\Omega^j(|x\rangle_d |y\rangle_2) = |x\rangle_d \otimes \Omega^{xj} |y\rangle_2 \tag{73}$$
$$= |x\rangle_d \otimes \omega^{xyj} |y\rangle_2 \tag{74}$$

where $|x\rangle_d$ is the control qudit and $|y\rangle_2$ the target qubit.

Claim 4.1. *In the Fourier basis, the C-Ω^j operator acts on the product state $|+_k\rangle_d \otimes |1\rangle_2$ to give $|+_{k+j}\rangle_d \otimes |1\rangle_2$*

Proof. The Fourier basis state $|+_k\rangle_d$ is given by

$$|+_k\rangle_d = \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{km} |m\rangle . \tag{75}$$

Therefore

$$C\text{-}\Omega^j \left(|+_k\rangle_d |1\rangle_2\right) = \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} C\text{-}\Omega^j \left(\omega^{km} |m\rangle_d \otimes |1\rangle_2\right) \tag{76}$$
$$= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{mj} \left(\omega^{km} |m\rangle_d \otimes |1\rangle_2\right) \tag{77}$$
$$= \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{(k+j)m} |m\rangle_d \otimes |1\rangle_2 \tag{78}$$
$$= \left(\frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \omega^{(k+j)m} |m\rangle_d\right) \otimes |1\rangle_2 \tag{79}$$
$$= |+_{k+j}\rangle_d \otimes |1\rangle_2 . \tag{80}$$

□

Claim 4.1 demonstrates the role of the index j, which essentially dictates in the Fourier basis how many states each Fourier basis state is shifted, and is analgous to the C-X operator in the computational basis. It also highlights the somewhat counterintuitive nature of the circuit, and indeed the counterintuitive way ancilla-based quantum computing can exploit quantum phenomena. A controlled operation is utilised, where the qudit is the control and the qubit is the target, yet it is the qudit state that is changed and the qubit is left 'untouched'. This idea is crucial for this proposal to achieve the desired outcomes, as we are required to shift the qudit states, but are unable to do so without being in the Fourier basis, where delocalised phase changes shift the Fourier basis states. Analogously to the QW protocol, the qudit can be thought of as a walker driven by the ancilla qubit 'coin'.

4.1 Transfer

Assume that we have two qudits in labs A and B which are spatially separated, and each lab shares a qubit each from an entanglement Bell pair. The composite state is therefore given by

$$|00\rangle_d^{(A,B)} \otimes \frac{1}{\sqrt{2}} \left(|00\rangle_2^{(A,B)} + |11\rangle_2^{(A,B)} \right). \tag{81}$$

Note that in the above expression, the qudits and qubits in both labs, A and B, are in fact in the same states. This is true for the entire running of the circuit and as such there is no need to explictly differentiate the qudits and ancilla qubits in either lab so the superscript labelling is dropped for the rest of this section.

The qudits are first Fourier transformed into the Fourier conjugate basis to give

$$|+_0+_0\rangle_d \otimes \frac{1}{\sqrt{2}} \left(|00\rangle_2 + |11\rangle_2 \right) \tag{82}$$

$$= \frac{1}{\sqrt{2}} \Big(|+_0+_0\rangle_d \otimes |00\rangle_2 + |+_0+_0\rangle_d \otimes |11\rangle_2 \Big). \tag{83}$$

The change of basis is neccessary because to have entanglement in our qudits, they must be in superposition of at least two states of non-zero amplitude. Since only amplitudes can be delocalised across the qudits and ancilla qubits, in order to give other qudit states a non-zero amplitude, it is necessary to be in a basis where the basis states are related to one another by differences in their amplitudes. The Fourier basis is one where the basis states are in equal superposition of the all the computational basis states but differ by the relative phases in the superposition. Hence the need to be in the Fourier basis when using the delocalised amplitudes to effect changes in the qudits. The index j on the Ω has an analogy to the QW protocol where the iteration number dictates how many steps need to be taken in the QW. Using the same line of reasoning as outlined in Claim 3.1, on the n^{th} iteration of the protocol, $j = 2^{n-1}$ so in this instance ($n = 1$) $j = 1$. Using Claim 4.1, acting C-Ω in both A and B gives the state

$$\frac{1}{\sqrt{2}} \left(|+_0+_0\rangle_d \otimes |00\rangle_2 + |+_1+_1\rangle_d \otimes |11\rangle_2 \right). \tag{84}$$

The qudits are then transformed back into the computational basis,

$$\frac{1}{\sqrt{2}}\left(|00\rangle_d \otimes |00\rangle_2 + |11\rangle_d \otimes |11\rangle_2\right). \tag{85}$$

The operator U is a correctional gate that essentially pairs all the even numbered computational qudit basis states with $|0\rangle_2$ and the odd numbered computational basis states with $|1\rangle_2$. (A more complete explanation for this is given in Section 4.2.) In this instance, U can be ignored since no corrections are needed, so C-X is directly applied, giving

$$\frac{1}{\sqrt{2}}\left(|00\rangle_d \otimes |00\rangle_2 + |11\rangle_d \otimes |00\rangle_2\right). \tag{86}$$

$$= \underbrace{\frac{1}{\sqrt{2}}\left(|00\rangle_d + |11\rangle_d\right)}_{\text{Bell state}} \otimes |00\rangle_2 . \tag{87}$$

As shown in equation 87, the qudits now effectively form a Bell pair and the qubits are no longer entangled - the entanglement has been transferred to the qudits.

4.2 Accumulation

Assume that one ebit has been transferred to our qudits and a second ebit is to be transferred. Taking the final state given by equation 87 and replacing the ancilla qubits with another entangled Bell pair gives

$$\frac{1}{2}\left(|00\rangle_d + |11\rangle_d\right) \otimes \left(|00\rangle_2 + |11\rangle_2\right). \tag{88}$$

Again, the qudits are transformed to the Fourier basis before the C-Ω^j gate. As noted in the discussion on C-Ω^j above, for this iteration ($n = 2$) $j = 2$, giving the state

$$\frac{1}{2}\Big[\left(|+_0+_0\rangle_d + |+_1+_1\rangle_d\right) \otimes |00\rangle_2 + \left(|+_2+_2\rangle_d + |+_3+_3\rangle_d\right) \otimes |11\rangle_2\Big]. \tag{89}$$

After doing transforming back to the computational basis, the total state becomes

$$\frac{1}{2}\Big[\left(|00\rangle_d + |11\rangle_d\right) \otimes |00\rangle_2 + \left(|22\rangle_d + |33\rangle_d\right) \otimes |11\rangle_2\Big]. \tag{90}$$

Unlike the previous example for $n = 1$, here the correctional operator U is needed so that the C-X will ensure all the qubit states are $|0\rangle_2$, allowing our qubit state to be completely separable from its associated qudit. In order for the C-X to do this, it must act as the identity on $|0\rangle_2$, implement an X operation on $|1\rangle_2$. Recall from equation 21 that since X is self inverse for $d = 2$, the control must be even when the target is $|0\rangle_2$ and odd when the target is $|1\rangle_2$. Hence, U must implement

$$|1\rangle \mapsto |2\rangle \tag{91}$$

$$|2\rangle \mapsto |1\rangle , \tag{92}$$

which then gives the state

$$\frac{1}{2}\Big[(|00\rangle_d + |22\rangle_d) \otimes |00\rangle_2 + (|11\rangle_d + |33\rangle_d) \otimes |11\rangle_2\Big]. \tag{93}$$

As stated in Section 4.1, U has paired even numbered qudit basis states with $|00\rangle_2$ and odd numbered qudit basis states with $|11\rangle_2$. Therefore after the C-X operation the state becomes

$$\frac{1}{2}(|00\rangle_d + |22\rangle_d + |11\rangle_d + |33\rangle_d) \otimes |00\rangle_2 . \tag{94}$$

The two qudits are now in an entangled state with a log negativity of 2, and there is again no entanglement in the qubits, so the entanglement has been perfectly transferred. This protocol can be repeated again and again with the sole change needed being the index j of C-Ω^j and the operator U. For $n > 2$ there is no unique choice of U, as long as odd numbered states less than 2^{n-1} and the even states greater than or equal to 2^{n-1} are swapped in some way.

4.3 Retrieval

Given that this is a circuit that solely utilises unitary transformations, retrieval of entangled Bell pairs is trivially done by running the circuit backwards. Furthermore, there is no requirement to use the same ancilla qubits to retrieve the entanglement. By the end of the circuit the ancilla qubits are in the $|0\rangle$ state, therefore any pair of ancilla qubits in the $|0\rangle$ state may be used to retrieve the entanglement.

4.4 Further uses of the circuit

Although the goal of this research was to design a protocol in a similar setting to the QW based protocol that outperformed the QW protocol, further probing of the proposed circuit showed that it had further uses beyond storage of entanglement. In this section, we consider a setting in a single lab that has many qubits and a qudit, and we wishes to store the quantum state of the qubits. By taking one half of the circuit given for the two lab setting and adding a gate U^{-1} that undoes the basis state map seen before, as shown in Figure 7, we can in fact store arbitrary qubit states in our qudits. Furthermore, in the case that we store two or more qubit states via this circuit then it is in fact possible to retrieve the qubit states in a different order to that which they were stored, simply by keep a record of order in which they were stored. Therefore, this circuit can be utilised in turning qudits into quantum random access memory.

4.4.1 Quantum Random Access Memory

The procedure for utilising the circuit as a quantum random access memory is as follows. Assume that there is a qudit in state $|0\rangle_d$ and two qubits, qubit 1 and qubit 2, in states $a|0\rangle_2 + b|1\rangle_2$ and $c|0\rangle_2 + d|1\rangle_2$ respectively. We first run the circuit with the qudit and qubit 1,

$$|0\rangle_d \otimes (a|0\rangle + b|1\rangle) \longrightarrow (a|0\rangle_d + b|1\rangle_d) \otimes |0\rangle_2 . \tag{95}$$

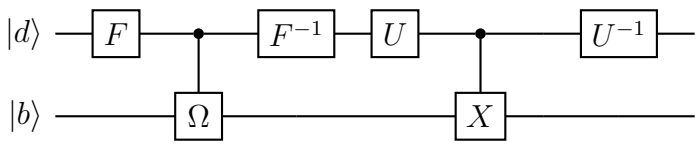


Figure 7: The AQC circuit with one qudit and qubit which can store arbitrary qubit states in the qudit.

We then replace qubit 1 with qubit 2 and run the circuit again,

$$(a\,|0\rangle_d + b\,|1\rangle_d) \otimes (c\,|0\rangle_2 + d\,|1\rangle_2) \longrightarrow (ac\,|0\rangle_d + bc\,|1\rangle_d + ad\,|2\rangle_d + bd\,|3\rangle_d) \otimes |0\rangle_2 \,. \tag{96}$$

In order to retrieve qubit 2 back, this can be done simply by running the circuit backwards. However, if we wish to retrieve the qubit 1 state then we first require a specific unitary operator. The form of the unitary transform can be found as follows. Rewrite each of the qudit basis state numbers in binary, i.e.

$$|0\rangle = |00\rangle \tag{97}$$
$$|1\rangle = |01\rangle \tag{98}$$
$$|2\rangle = |10\rangle \tag{99}$$
$$|3\rangle = |11\rangle \,. \tag{100}$$

Rewriting the final state of equation 96 in this way we obtain the following expression

$$ac\,|0\rangle + bc\,|1\rangle + ad\,|2\rangle + bd\,|3\rangle = ac\,|00\rangle + bc\,|01\rangle + ad\,|10\rangle + bd\,|11\rangle \,, \tag{101}$$

which can be rewritten as the product state

$$\underbrace{(c\,|0\rangle + d\,|1\rangle)}_{\text{Qubit 2}} \otimes \underbrace{(a\,|0\rangle + b\,|1\rangle)}_{\text{Qubit 1}} . \tag{102}$$

Quite remarkably the qudit has an intuitive alternative expression as the product state of qubit 1 and qubit 2. If the circuit is run backwards, we will retrieve qubit 2. Therefore, to retrieve qubit 1 instead, the qudit state should be expressable as

$$\underbrace{(a\,|0\rangle + b\,|1\rangle)}_{\text{Qubit 1}} \otimes \underbrace{(c\,|0\rangle + d\,|1\rangle)}_{\text{Qubit 2}} . \tag{103}$$

This can be thought of as switching the positions of our two qubits, which leads us to the unitary transformation we need. We need a map, M, which will switch the positions of our two qubits. M is found by mapping each of the binary qudit state representations $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ to the basis state with the binary digits switched.

$$|00\rangle \mapsto |00\rangle \tag{104}$$
$$|01\rangle \mapsto |10\rangle \tag{105}$$
$$|10\rangle \mapsto |01\rangle \tag{106}$$
$$|11\rangle \mapsto |00\rangle \,. \tag{107}$$

We can check that this gives us the form that we want taking the RHS of equation 101 and acting M on it to obtain

$$M\left(ac\left|00\right\rangle + ad\left|01\right\rangle + bc\left|10\right\rangle + bd\left|11\right\rangle\right) = ac\left|00\right\rangle + ad\left|10\right\rangle + bc\left|01\right\rangle + bd\left|11\right\rangle \tag{108}$$

$$= \underbrace{\left(a\left|0\right\rangle + b\left|1\right\rangle\right)}_{\text{Qubit 1}} \otimes \underbrace{\left(c\left|0\right\rangle + d\left|1\right\rangle\right)}_{\text{Qubit 2}}, \tag{109}$$

as required. M can be expressed in terms of the original qudit state labelling by converting the binary back

$$\left|0\right\rangle \mapsto \left|0\right\rangle \tag{110}$$

$$\left|1\right\rangle \mapsto \left|2\right\rangle \tag{111}$$

$$\left|2\right\rangle \mapsto \left|1\right\rangle \tag{112}$$

$$\left|3\right\rangle \mapsto \left|3\right\rangle. \tag{113}$$

The matrix representation of M is given by

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{114}$$

Generalising this to larger numbers of qubits stored is done via the same thought process. Taking the example of storing three qubits,

$$\begin{aligned} &\text{Qubit 1 } a\left|0\right\rangle + b\left|1\right\rangle \\ &\text{Qubit 2 } c\left|0\right\rangle + d\left|1\right\rangle \\ &\text{Qubit 3 } e\left|0\right\rangle + f\left|1\right\rangle. \end{aligned}$$

When stored in the order above, the qudit state is given by

$$ace\left|0\right\rangle + ade\left|1\right\rangle + bce\left|2\right\rangle + bde\left|3\right\rangle + ace\left|4\right\rangle + adf\left|5\right\rangle + bcf\left|6\right\rangle + bdf\left|7\right\rangle, \tag{115}$$

which again, when converted to binary numbers, can be expressed as the product state

$$\left(e\left|0\right\rangle + f\left|1\right\rangle\right) \otimes \left(c\left|0\right\rangle + d\left|1\right\rangle\right) \otimes \left(a\left|0\right\rangle + b\left|1\right\rangle\right). \tag{116}$$

If we want to retrive qubit 1 then we require M to perform the following map

$$\left|1\right\rangle = \left|001\right\rangle \mapsto \left|100\right\rangle = \left|4\right\rangle \tag{117}$$

$$\left|3\right\rangle = \left|011\right\rangle \mapsto \left|110\right\rangle = \left|6\right\rangle \tag{118}$$

$$\left|4\right\rangle = \left|100\right\rangle \mapsto \left|001\right\rangle = \left|1\right\rangle \tag{119}$$

$$\left|6\right\rangle = \left|110\right\rangle \mapsto \left|011\right\rangle = \left|3\right\rangle \tag{120}$$

with all other basis states mapped to themselves as they are identical when switching the first and last digits of their binary representations. Having operated M on the qudit state,

we can now run the circuit backwards in order to retrieve qubit 1. As with retrieving Bell pairs, any qubit in the state $|0\rangle$ can be used to retrieve qubit 1.

In a practical setting it may be more efficient to build a gate, P, that permutes the order of the qubits instead of swapping two digits. One example of P would be the map that permutes each qubit one place the right,

$$|1\rangle = |001\rangle \mapsto |100\rangle = |4\rangle \tag{121}$$
$$|2\rangle = |010\rangle \mapsto |001\rangle = |1\rangle \tag{122}$$
$$|3\rangle = |011\rangle \mapsto |101\rangle = |5\rangle \tag{123}$$
$$|4\rangle = |100\rangle \mapsto |010\rangle = |2\rangle \tag{124}$$
$$|5\rangle = |101\rangle \mapsto |110\rangle = |6\rangle \tag{125}$$
$$|6\rangle = |110\rangle \mapsto |011\rangle = |3\rangle . \tag{126}$$

Again $|0\rangle , |7\rangle$ are mapped to themselves as the permutation does not affect them. In this way we would be able to retrieve any of the qubits by continually applying P until the qubit we wish to retrieve is in the right 'place'. Utilising a single gate would also be more appropriate for the ancilla-based quantum computing setting where we wish to have a minimal gate set applied to the control qudits. However, this would require multiple applications of the same gate which, without adequate error correction, would lead to greater decoherence of the quantum information than if just a single switching gate M were applied.

5 Discussion

5.1 Further Steps

As discussed in Section 4.4, the circuit given in Figure 6 has potential uses beyond the original scope in which is has been designed. It would be of interest to see if it can also be applied to a scenario where we wish to transfer multipartite entanglement to higher dimensional qudits, for example transferring GHZ states or W states. Such analysis should be relatively straightforward to generalise to the multipartite scenario, although extending computational simulations may not be so trivial without adequate computational power. Furthermore, the analysis of this work could be brought to greater completion if the circuit was generalised to the scenario where we wish to store qudit states rather than just qubit states.

5.2 Conclusions

Acknowledgements

The author would like to thank...

References

[1] Charles H. Bennett and Stephen J. Wiesner. 'Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states'. In: *Phys. Rev. Lett.* 69 (20 Nov. 1992), pp. 2881–2884. DOI: 10.1103/PhysRevLett.69.2881. URL: https://link.aps.org/doi/10.1103/PhysRevLett.69.2881.

[2] Charles H. Bennett and Gilles Brassard. 'Quantum cryptography: Public key distribution and coin tossing'. In: vol. 560. Jan. 1984, pp. 175–179. DOI: 10.1016/j.tcs.2011.08.039.

[3] Charles H. Bennett et al. 'Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels'. In: *Phys. Rev. Lett.* 70 (13 Mar. 1993), pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.1895.

[4] Neil Shenvi, Julia Kempe, and K. Birgitta Whaley. 'Quantum random-walk search algorithm'. In: *Phys. Rev. A* 67 (5 May 2003), p. 052307. DOI: 10.1103/PhysRevA.67.052307. URL: https://link.aps.org/doi/10.1103/PhysRevA.67.052307.

[5] Andrew M. Childs. 'Universal Computation by Quantum Walk'. In: *Physical Review Letters* 102.18 (May 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.102.180501. URL: http://dx.doi.org/10.1103/PhysRevLett.102.180501.

[6] Viv Kendon. 'Decoherence in quantum walks – a review'. In: *Mathematical Structures in Computer Science* 17.06 (Nov. 2007). ISSN: 1469-8072. DOI: 10.1017/s0960129507006354. URL: http://dx.doi.org/10.1017/S0960129507006354.

[7] Andrew M. Childs et al. 'Exponential algorithmic speedup by a quantum walk'. In: *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC '03* (2003). DOI: 10.1145/780542.780552. URL: http://dx.doi.org/10.1145/780542.780552.

[8] Taira Giordani et al. 'Entanglement transfer, accumulation and retrieval via quantum-walk-based qubit-qudit dynamics'. In: *arXiv e-prints* (Oct. 2020). arXiv: 2010.07127 [quant-ph]. URL: https://arxiv.org/abs/2010.07127.

[9] Martin B. Plenio and Shashank Virmani. 'An Introduction to entanglement measures'. In: *Quant. Inf. Comput.* 7 (2007), pp. 1–51. arXiv: quant-ph/0504163.

[10] Ben Tregenna et al. 'Controlling discrete quantum walks: coins and initial states'. In: *New Journal of Physics* 5 (July 2003), pp. 83–83. ISSN: 1367-2630. DOI: 10.1088/1367-2630/5/1/383. URL: http://dx.doi.org/10.1088/1367-2630/5/1/383.

Appendices

A Equivalence of Maximally Entangled Ququart State to 2 Bell States

Here we present a proof of the no-cloning theorem.

Proof. Let $\mathcal{H}$ be a Hilbert space with two quantum systems A and B. Assume there exists a unitary operation $U \in \mathcal{H}$ that can clone any arbitrary $|\psi\rangle_A$ onto any arbitrary $|\phi\rangle_B$.

$$U|\psi\rangle \otimes |\phi\rangle = |\psi\rangle \otimes |\psi\rangle \tag{A.1}$$
$$\implies U^{-1}|\psi\rangle \otimes |\psi\rangle = |\psi\rangle \otimes |\phi\rangle \tag{A.2}$$
$$\implies U^{\dagger}|\psi\rangle \otimes |\psi\rangle = |\psi\rangle \otimes |\phi\rangle \tag{A.3}$$

Suppose we have any two arbitrary $|\psi\rangle_A, |\tilde{\psi}\rangle_A$ s.t. $0 \leq \langle\psi|\tilde{\psi}\rangle \leq 1$.

$$\langle\psi|\tilde{\psi}\rangle = \langle\psi|\tilde{\psi}\rangle\langle\phi|\phi\rangle \tag{A.4}$$
$$= (\langle\psi| \otimes \langle\phi|)(|\tilde{\psi}\rangle \otimes |\phi\rangle) \tag{A.5}$$
$$= (\langle\psi| \otimes \langle\psi|)UU^{\dagger}(|\tilde{\psi}\rangle \otimes |\tilde{\psi}\rangle) \tag{A.6}$$
$$= (\langle\psi| \otimes \langle\psi|)(|\tilde{\psi}\rangle \otimes |\tilde{\psi}\rangle) \tag{A.7}$$
$$= \langle\psi|\tilde{\psi}\rangle^2 \tag{A.8}$$
$$\implies \langle\psi|\tilde{\psi}\rangle = 0 \text{ or } \langle\psi|\tilde{\psi}\rangle = 1 \tag{A.9}$$

which is a contradiction. □