Quantum Information Processing, Vol. 5, No. 5, October 2006 (© 2006)
DOI: 10.1007/s11128-006-0028-z

Programmable Quantum Processors*

Vladimír Bužek,[1,2,3,6] Mark Hillery,[4] Mário Ziman,[1,3,5] and Marián Roško[1,3]

Received May 12, 2005; accepted December 21, 2005; Published online July 12, 2006

A quantum processor is a device with a data register and a program register. The input to the program register determines the operation, which is a completely positive linear map, that will be performed on the state in the data register. We develop a mathematical description for these devices. We generalize the concept of quantum programmable processors and we propose programmable measurement devices.

KEY WORDS: Quantum program registers; programmable quantum processor; positive operator-valued measurements.

PACS: 03.67.-a; 03.67.Lx; 03.65.Ta.

1. INTRODUCTION

The coherent control of individual quantum systems is one of the most exciting achievements in physics in the last decade.(1) The possibility of controlling quantum dynamics has far reaching consequences for quantum technologies, in particular, for quantum computing.(2) One of the best-known applications of coherent control in quantum physics is the state preparation of an *individual* quantum system. For example, a particular state of the vibrational motion of a trapped ion can be prepared by using

*We dedicate this paper to Anton Zeilinger on the occasion of his 60th birthday.

[1]Research Center for Quantum Information, Institute of Physics, Slovak Academy of Sciences, Dúbravská cesta 9, 845 11 Bratislava, Slovakia.
[2]Abteilung für Quantenphysik, Universität Ulm, 89069 Ulm, Germany.
[3]*Quniverse*, Líščie údolie 116, 841 04 Bratislava, Slovakia.
[4]Department of Physics, Hunter College of CUNY, 695 Park Avenue, New York, NY 10021, USA.
[5]Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic.
[6]To whom correspondence should be addressed. E-mail: buzek@savba.sk

a well-defined sequence of external laser pulses. Another possibility is to focus on controlling the dynamics, that is, the unitary evolution operator. One way of doing this is to realize a particular evolution operator by means of a sequence of "elementary" interactions, which are sequentially turned on and off (for more details see Refs. 3–5 and for a specific application to trapped-ions see Ref. 6 and references therein).

In the theory of quantum coherent control it is assumed that the control of the dynamics is realized via external *classical* parameters, such as the intensity of a laser pulse or the duration of an interaction (see e.g., Refs. 3, 4). In this case, the information that controls the quantum system is classical, and it is set by an experimentalist to achieve a single, fixed outcome. This is analogous to programming a computer (processor) to perform a single task by setting dials or switches to particular positions, each task requiring different positions.

In the present paper we will review a different type of quantum control. We will assume that the information about the quantum dynamics of the system under consideration is not represented by classical external parameters, but rather is encoded in the state of another quantum system. A typical example of such an arrangement is a controlled-NOT (C-NOT) operation (or, in general, a controlled-U operation). In this case, the specific operation performed on the system, the target, depends on the state of a second quantum system, the control. If the control qubit is in the state $|0\rangle$ the target qubit is left unchanged, but if it is in the state $|1\rangle$, then a NOT operation is applied to the target qubit. This means that this device can perform at least two operations on the target qubit, the identity and NOT. There are, however, further possibilities. Let us suppose that the control qubit is initially in a superposition of $|0\rangle$ and $|1\rangle$, and that we are only interested in the target qubit at the output of the device, so that we trace out the control qubit to obtain the reduced density matrix of the target qubit. The action of the C-NOT gate on the target qubit can then be described as a completely positive, linear map acting on the initial density matrix of the target qubit, with the actual map being determined by the state of the control qubit. We take this device to be a model for a programmable quantum gate array, or quantum processor. Generally speaking, a programmable quantum processor (see Fig. 1) is a device that implements a completely positive linear map, which is determined by the state of one quantum system, on a second quantum system. These processors have two registers, the data register and the program register. The data register contains the quantum system on which the map is going to be applied, and the program register contains the quantum system whose state determines the map. The third element of this device is a fixed array of quantum gates that act on both the program and the data state. The

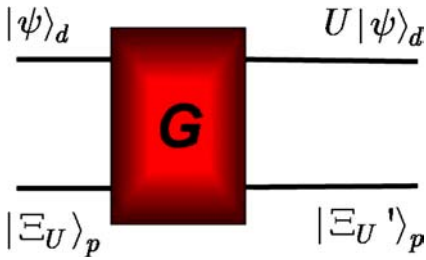

Fig. 1. A model of a general quantum processor G that implements a unitary operation U on the data register.

virtue of this arrangement is that we do not have to build a different processor every time we want to realize a new map, we simply change the program state. This allows us greater flexibility than a device in which the map is determined by setting external parameters. For example, it could be the case that we do not even know what the program state is. This would occur when the state of the program register is the output of another quantum device. We will refer to the selection of the program state to perform a desired operation as *quantum programming*.

Programmable quantum processors (gate arrays) were first considered by Nielsen and Chuang.[7] They were only interested in the case in which a unitary operation, rather than a more general completely positive linear map, is performed on the state in the data register. If $|\psi\rangle_d$ is the state of the data register, $|\Xi_U\rangle_p$ a program state that implements the operator U on the data state, and G the overall unitary operation implemented by the fixed gate array, then their processor carries out the transformation (see Fig. 1)

$$G(|\psi\rangle_d \otimes |\Xi_U\rangle_p) = U|\psi\rangle_d \otimes |\Xi'_{U,\psi}\rangle_p, \tag{1.1}$$

where $|\Xi'_{U,\psi}\rangle_p$ is the state of the program register after the transformation G has been carried out. The subscripts U and ψ indicate that this state can depend on both the operation U and the state $|\psi\rangle_d$ of the data register d. Nielsen and Chuang were able to prove a number of results about this device. First, they showed that the output of the program register does not depend on the data register, a fact that follows from the unitarity of G. Second, they proved that the number of possible programs is equal to the dimension of the program register.

In order to become more familiar with the concept of programmable processors let us assume the first of these results and show how to prove the second. Consider two program states, $|\Xi_U\rangle_p$ and $|\Xi_V\rangle_p$, that cause the

operators U and V, respectively, to act on the data register. This implies that

$$\begin{aligned} G(|\psi\rangle_d \otimes |\Xi_U\rangle_p) &= U|\psi\rangle_d \otimes |\Xi'_U\rangle_p\,; \\ G(|\psi\rangle_d \otimes |\Xi_V\rangle_p) &= V|\psi\rangle_d \otimes |\Xi'_V\rangle_p. \end{aligned} \tag{1.2}$$

The unitarity of G implies that

$${}_p\langle \Xi_V|\Xi_U\rangle_p = {}_d\langle\psi|V^{-1}U|\psi\rangle_d\, {}_p\langle \Xi'_V|\Xi'_U\rangle_p, \tag{1.3}$$

and if ${}_p\langle \Xi'_V|\Xi'_U\rangle_p \neq 0$, then

$${}_d\langle\psi|V^{-1}U|\psi\rangle_d = \frac{{}_p\langle \Xi_V|\Xi_U\rangle_p}{{}_p\langle \Xi'_V|\Xi'_U\rangle_p}. \tag{1.4}$$

The left-hand side of this equation depends on $|\psi\rangle_d$ while the right does not. The only way this can be true is if

$$V^{-1}U = e^{i\phi} I, \tag{1.5}$$

for some real ϕ. This means that the operators U and V are the same up to a phase. If we want these operators to be different, we must have that ${}_p\langle \Xi'_V|\Xi'_U\rangle_p = 0$, which by Eq. (1.3) implies that ${}_p\langle \Xi_V|\Xi_U\rangle_p = 0$. Therefore, the program states corresponding to different unitary operators must be orthogonal. This implies that the dimension of the program register must be greater than or equal to the number of different unitary operators that can be performed on the data register.

From the investigation of Nielsen and Chuang[7] it follows that a deterministic *universal* quantum processor of finite size does not exist. The problem is that a new dimension must be added to the program space for each unitary operator U that one wants to be able to perform on the data $|\psi\rangle_d$. A similar situation holds if one studies quantum circuits that implement completely-positive, trace-preserving maps rather than just unitary operators.[8,9] Some families of maps can be implemented with a finite program space, for example, the phase damping channel, but others, such as the amplitude damping channel, require an infinite program space. If one drops the requirement that the processor is deterministic, then universal processors become possible.[7,10–12] These processors are probabilistic: they sometimes fail, but we know when this happens.

In a probabilistic processor we demand that by a measurement of the program register, we can tell whether the desired unitary operation has been performed on the data state or whether some other unitary operation has been performed upon it, i.e., that the state of the program register associated with the execution of U, $|\Xi'_U\rangle_p$, is orthogonal to the states

of the program register associated with other, undesired, outcomes on the data state (the identity of these states of the program register will in general be dependent on the nature of the processor itself). A model of this is shown in Fig. 2 where the outcome of the measurement of the program register, $|k\rangle_p$ indicates which unitary operation, U_k, has been performed on the data state.

The simplest case of desired programmable operation on a qubit is the execution of a $U(1)$ transformation, $U(\theta) = e^{i\theta\sigma_z/2}$, upon a data qubit $|\psi\rangle_d = \alpha|0\rangle_d + \beta|1\rangle_d$. Here the *unknown* phase of the rotation θ is encoded in the programme state

$$|\Xi_\theta\rangle_p = \frac{1}{\sqrt{2}}\left(|0\rangle_p + e^{-i\theta}|1\rangle_p\right), \tag{1.6}$$

while the processor itself is represented by a C-NOT gate with data qubit as control and program qubit as target, followed by a measurement of the program qubit in the basis $\{|0\rangle_p, |1\rangle_p\}$ (see Fig. 3).
The action of the C-NOT processor on the data and the program input states is

$$\begin{aligned} |\psi\rangle_d|\Xi_\theta\rangle_p \longrightarrow & \frac{1}{\sqrt{2}}U(\theta)|\psi\rangle_d|0\rangle_p \\ & + \frac{1}{\sqrt{2}}U(-\theta)|\psi\rangle_d|1\rangle_p . \end{aligned} \tag{1.7}$$

From this equation we see that when a projective measurement in the computer basis $\{|0\rangle, |1\rangle\}$ on the program qubit at the output of the C-NOT is performed and the result $|0\rangle$ is registered then the data qubit that has been prepared in an unknown state $|\psi\rangle$ is rotated by the *unknown* angle θ as desired, i.e., with probability 1/2 we obtain the state $U(\theta)|\psi\rangle_d$ (see Fig. 4). On the other hand, when the program qubit is measured in the

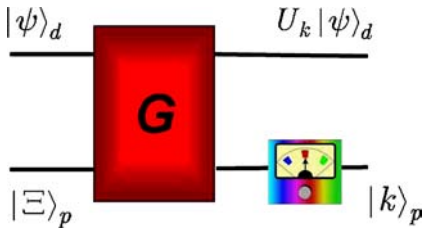

Fig. 2. A model of a probabilistic general quantum processor. On the output of the program register a measurement is performed.

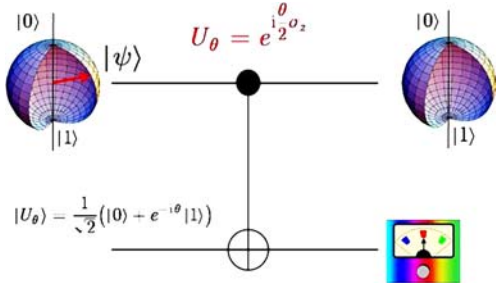

Fig. 3. A model of a probabilistic C-NOT quantum processor performing the $U(1)$ rotation of the input data state $|\psi\rangle$ by the angle θ that is encoded in the program state $|\Xi_\theta\rangle$ given by Eq. (1.6). On the output of the program register a measurement is performed. A state space of the data register is represented by the Bloch sphere.

state $|1\rangle_p$ then the data qubit is rotated in the opposite ("wrong") direction, i.e., with probability 1/2 we obtain at the output of the probabilistic processor the state $U(-\theta)|\psi\rangle_d$ (see Fig. 5).

1.1. Applications of Programmable Quantum Processors

Before we proceed, we would like to justify the concept of a programmable quantum processor. One may consider several arguments why programmable quantum processors might be of interest. The most important

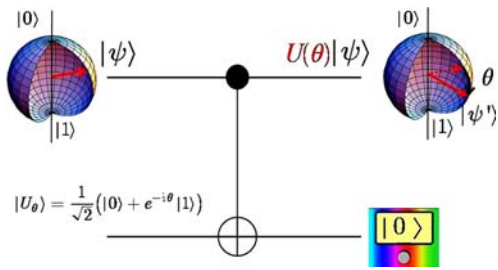

Fig. 4. A model of a probabilistic C-NOT quantum processor performing the $U(1)$ rotation of the input data state $|\psi\rangle$ by the angle θ that is encoded in the program state $|\Xi_\theta\rangle$. When the measurement performed on the program qubit result in the state $|0\rangle_p$ the desired rotation $U(\theta)$ is performed on the data qubit. The probability of success is equal to 1/2.

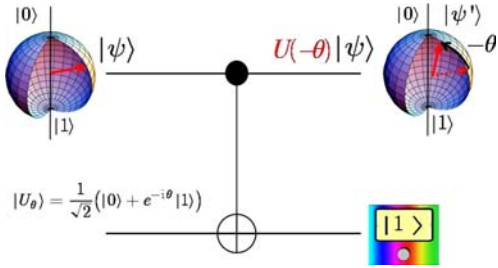

Fig. 5. A model of a probabilistic C-NOT quantum processor performing the $U(1)$ rotation of the input data state $|\psi\rangle$ by the angle θ that is encoded in the program state $|\Xi_\theta\rangle$. When the measurement performed on the program qubit results in the state $|1\rangle_p$ the rotation $U(-\theta)$ in the wrong direction is performed on the data qubit. The probability of this result is equal to 1/2.

argument is as follows: Let us imagine a situation when a set of instructions that characterize an operation to be performed on the data is encoded in a *single* copy of a quantum system. This may happen when the set of instructions (a program) is obtained as an output of a quantum computer (whatever this device is). This output state might be in general *unknown*. In this situation one has two options: Firstly, one can measure and estimate the program state[13–16] and with the obtained classical information one can perform a *classical* control of the evolution of the data register. The main obstacle in this approach is that the fidelity of estimation of a state of quantum system based on a measurement of just a single copy of the state is negligibls small (it is inversely proportional to a dimension of the Hilbert space of the program register[13–18]). This is the reason why the programmable quantum processor that takes as an input the unknown quantum program register is a better alternative. The quantum processor will perform operations that are specified by the program register even though a (classical) user of the processor does not have an information about the set of instructions. Another aspect of the encoding of quantum operations in the states of program registers has been discussed by Huelga and coworkers.[19] In this paper the implementation of an arbitrary unitary operation U upon a distant quantum system has been considered. This so called teleportation of unitary operations has been formally represented as a completely positive, linear, trace preserving map on the set of density operators of the program and data registers:

$$\mathcal{T}\left[|\xi\rangle_{ab} \otimes |\Xi_U\rangle_p \otimes |\psi\rangle_d\right] = |\tilde{\xi}_U\rangle_{ap} \otimes (U|\psi\rangle_d) \tag{1.8}$$

Here $|\xi\rangle_a$ represents a specific entangled state that is shared by two parties, Alice and Bob, who want to teleport the unitary operation U from Alice to Bob. Huelga et al.[19] have investigated protocols which achieve the teleportation of U using local operations, classical communication and shared entanglement.

1.2. Content of the Paper

In this paper we will present a concept of programmable quantum processors. We will study and classify deterministic processors that realize completely positive linear maps (Sec. 2). In Sec. 3 we introduce probabilistic programmable processors and we analyze in detail what type of operations can be realized with these processors. We will show that the so-called quantum information distributor (QID) is particularly useful realization of universal probabilistic programmable processor for qudits. In Sec. 4 we will show how the probability of success of probabilistic programmable processors can be increased by using more sophisticated programs. We will continue this discussion in Sect. 5 where we will consider an interesting case with N copies of the same program. Section 6 is devoted to a problem of processor design. Specifically, we will analyze how to design ("construct") a processor that is supposed to implement a given set of operations. In this section we will also discuss briefly a problem of quantum simulations. In Sec. 7 we will show that programmable processors can be used to implement generalized Positive operator valued measures (POVM) measurements. It is not always necessary to perform a desired operation on the data register perfectly. Therefore in Sec. 8 we address the problem of approximate processors. Finally, in Sec. 9 we briefly summarize our results.

2. DETERMINISTIC QUANTUM PROCESSORS: PROPERTIES AND CLASSIFICATIONS

As we have already mentioned in the Introduction a general programmable quantum processor consists of two registers, a data register and a program register, and a fixed array of quantum gates. The input state that goes into the program register encodes an operation we want to perform on the data register. We would first like to show that the action of the processor can be fully described by a specific set of linear operators.

2.1. Quantum States of Program Register

First we discuss the role of the program states in the concept of programmable quantum processors. In particular, we will investigate how the transformations that are supposed to be executed on the data can be encoded in quantum states of program registers.

2.1.1. Pure States

Let $|\psi\rangle_d$ be the input state of the data register, $|\Xi\rangle_p$ be the input program state and G be the unitary operator that describes the action of the array of quantum gates. If $\{|j\rangle_p | j = 1, \ldots N\}$ is a basis for the space of program states, then we have that

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = \sum_{j=1}^{N} |j\rangle_p \, {}_p\langle j|G(|\psi\rangle_d \otimes |\Xi\rangle_p). \tag{2.1}$$

If we define the operator $A_j(\Xi)$, which acts on the data register, by

$$A_j(\Xi)|\psi\rangle_d = {}_p\langle j|G(|\psi\rangle_d \otimes |\Xi\rangle_p), \tag{2.2}$$

then we have that

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = \sum_{j=1}^{N} A_j(\Xi)|\psi\rangle_d \otimes |j\rangle_p. \tag{2.3}$$

This means that the output density matrix of the data register is given by

$$\rho_d^{\text{out}} = \sum_{j=1}^{N} A_j(\Xi)|\psi\rangle_d \, {}_d\langle\psi|A_j^{\dagger}(\Xi). \tag{2.4}$$

The operator $A_j(\Xi)$ depends on the program state, but it can be expressed in terms of operators that do not. Define the operators

$$A_{jk} = A_j(|k\rangle) = {}_p\langle j|G|k\rangle_p, \tag{2.5}$$

where $|k\rangle$ is one of the basis states we have chosen for the space of program states. We have that for any program state $|\Xi\rangle$

$$A_j(\Xi) = \sum_{k=1}^{N} {}_p\langle k|\Xi\rangle_p A_{jk}. \tag{2.6}$$

This means that the operators A_{jk} completely characterize the processor. We shall call these operators the basis operators for the processor. These operators have the following property,

$$\sum_{j=1}^{N} A_{jk_1}^{\dagger} A_{jk_2} = \sum_{j=1}^{N} \langle k_1 | G^{\dagger} | j \rangle \langle j | G | k_2 \rangle = I_d \delta_{k_1 k_2}, \tag{2.7}$$

where we have used the decomposition $\sum_j |j\rangle\langle j| = I_p$.

An obvious question to ask at this point is whether any set of operators satisfying Eq. (2.7) corresponds to a quantum processor. The following construction allows us to show that this is the case.[20] Given a set of N^2 operators acting on $\mathcal{H}_d$, we can construct an operator, G, acting on the product space $\mathcal{H}_d \otimes \mathcal{H}_p$, where $\mathcal{H}_p$ is an N-dimensional space with basis $\{|k\rangle_p | k = 1, \ldots N\}$. We set

$$G = \sum_{j,k=1}^{N} A_{jk} \otimes |j\rangle_p \, {}_p\langle k|. \tag{2.8}$$

It is now necessary to verify that G constructed in this way is unitary. Noting that

$$G^{\dagger} = \sum_{j,k=1}^{N} A_{jk}^{\dagger} \otimes |k\rangle_p \, {}_p\langle j|, \tag{2.9}$$

we see that Eq. (2.7) implies that $G^{\dagger}G = GG^{\dagger} = I$, so that G preserves the length of vectors and is unitary.

It is possible to express the basis operators for closely related processors in terms of each other. For example, if $\{B_{jk} | j, k = 1, \ldots N\}$ are the basis operators for $G^{\dagger}$, then from Eq. (2.9) we see that $B_{jk} = A_{kj}^{\dagger}$. If G_1 and G_2 are two processors (unitary operators) with basis operators $\{A_{jk}^{(1)} | j, k = 1, \ldots N\}$ and $\{A_{jk}^{(2)} | j, k = 1, \ldots N\}$, respectively, then the basis operators, C_{jk}, for the processor corresponding to the operator $G_1 G_2$ are

$$C_{jk} = \sum_{n=1}^{N} A_{jn}^{(1)} A_{nk}^{(2)}. \tag{2.10}$$

This follows immediately if both G_1 and G_2 are expressed in the form given in Eq. (2.8) and then multiplied together. If we apply this equation to the case $G_1 = G$ and $G_2 = G^{\dagger}$, and note that $GG^{\dagger} = I$, we have that

$$\sum_{j=1}^{N} A_{k_1 j} A_{k_2 j}^{\dagger} = I_d \delta_{k_1 k_2}. \tag{2.11}$$

It is clearly possible to generalize Eq. (2.10) to the case when there is a product of more than two operators.

2.1.2. General Program States

Let us suppose the program is represented by a mixed state $\varrho_p = \sum_{kl} R_{kl}|k\rangle\langle l|$. Then for the induced mapping we have

$$\begin{aligned}\varrho_d^{\text{out}} &= \sum_{klmn} R_{kl} A_{mk} \varrho_d^{\text{in}} A_{nl}^{\dagger} \text{Tr}_p(|m\rangle_p\langle n|) \\ &= \sum_{klm} R_{kl} A_{mk} \varrho_d A_{ml}^{\dagger}. \end{aligned} \tag{2.12}$$

We shall denote by $\mathcal{C}_G$ the set of completely positive linear maps realizable by using the fixed processor G and any mixed state in the program space as a program.

Let us now address the question of whether it is possible to find a second processor, G', that can realize any map in the set $\mathcal{C}_G$ using only pure state programs. Any mixed state in $\mathcal{H}_p$ can be purified, but the purification is not unique.[2,21] We begin by defining a new program space, $\mathcal{H}_{p'}$ and choosing the purification of the density operator represented in terms of its spectral decomposition as $\varrho_p = \sum_k \lambda_k |\chi_k\rangle\langle\chi_k|$ in the following way

$$\varrho_p \longrightarrow |\Phi\rangle_{p'} = \sum_k \sqrt{\lambda_k}|\chi_k\rangle_p \otimes |k\rangle\,. \tag{2.13}$$

We define the unitary operator corresponding to the new processor, which acts on the space $\mathcal{H}_d \otimes \mathcal{H}_{p'}$, by

$$G' := G \otimes I\,. \tag{2.14}$$

The conjecture is that processor G' with the pure program state $|\Phi\rangle_{p'}$ will produce the same mapping as the processor G with the mixed program state ϱ_p. If this is true, then we will have shown that by using only pure program states with the processor G', we can implement the entire class of superoperators $\mathcal{C}_G$.

In order to prove this we have to show that

$$\text{Tr}_p G \varrho_d \otimes \varrho_p G^{\dagger} = \text{Tr}_{p'} G' \varrho_d \otimes \varrho_{p'} G'^{\dagger} \tag{2.15}$$

for all ϱ_d. The right-hand side of this equation can be rewritten as

$$\begin{aligned} & T_{p'} G' \varrho_d \otimes \varrho_{p'} G'^{\dagger} \\ & = \mathrm{Tr}_{p'} \left[\sum_{kl} \sqrt{\lambda_k \lambda_l} \left(G \varrho_d \otimes |\chi_k\rangle\langle\chi_l| G^{\dagger} \right) \otimes |k\rangle\langle l| \right] \\ & = \sum_{kl} \sqrt{\lambda_k \lambda_l} \mathrm{Tr}_p \left[\left(G \varrho_d \otimes |\chi_k\rangle\langle\chi_l| G^{\dagger} \right) \delta_{kl} \right] \\ & = \mathrm{Tr}_p \left[G \varrho_d \otimes \left(\sum_k \lambda_k |\chi_k\rangle\langle\chi_k| \right) G^{\dagger} \right] \\ & = \mathrm{Tr}_p G \varrho_d \otimes \varrho_p G^{\dagger}, \end{aligned} \tag{2.16}$$

which proves Eq. (2.15). Therefore, we can conclude that it is possible to "mimic" mixed program states for a given processor by introducing a larger program space $\mathcal{H}_{p'}$ and a new processor mapping $G' = G \otimes I$.

2.1.3. Correspondence Between Programs and Mappings

We have just seen that two different programs on two different processors can lead to the same mapping, and now we would like to examine whether different programs on the same processor can produce identical mappings. We shall illustrate this possibility by means of a simple example. Let Q_p be a projection operator on the program space whose range has dimension D, where $1 < D < N$, and let U_1 and U_2 be two different unitary operators on the data space. Consider the processor given by

$$G = U_1 \otimes Q_p + U_2 \otimes (I_p - Q_p). \tag{2.17}$$

Any program state in the range of Q_p produces the mapping U_1 on the data state, and there are clearly an infinite number of these. Therefore, we can conclude that there are processors for which many program states produce the same operation on the data state.

We shall now show that the opposite can also occur, i.e., that there exists a processor, for which every program state (mixed or pure) encodes a different superoperator. To do so, we utilize results of Refs. 22 and 23 where the unitary transformation

$$G = \cos\phi I + i \sin\phi S \tag{2.18}$$

was introduced. The swap operator $S = \sum_{kl} |kl\rangle\langle lk|$ is defined in any dimension. The so-called *partial swap* transformation G acts on two qudits

(d dimensional systems). Let us restrict our attention to qubits, and identify one of the qubits with the data register and other with the program system. In Ref. 23 it was shown that if the program system is prepared in the state $\varrho_p \equiv \xi$, then the induced map (superoperator) T_ξ is contractive with its fixed point equal to ξ. Since each contractive superoperator has only a single fixed point, we can conclude that different program states $\xi \neq \xi'$ induce different superoperators, i.e., $T_\xi \neq T_{\xi'}$. As a result we can conclude that in the processor given by Eq. (2.18), for any value of the parameter ϕ, the correspondence between programs and induced mappings is one-to-one. Finally, we note that the results presented in this paragraph also hold for qudits.

2.1.4. Equivalent Processors

We shall regard two processors G_1 and G_2 as equivalent if one can be converted into the other by inserting *fixed* unitary gate arrays at the input and output of the program register, that is if

$$G_2 = (I_d \otimes U_{p1}) G_1 (I_d \otimes U_{p2}), \tag{2.19}$$

where U_{p1} and U_{p2} are unitary transformations on the program space. If this equation is satisfied, then the processors defined by the two gate arrays will perform the same set of operations on data states, but the program states required to perform a given operation are different, and the outputs of the program registers will be different as well. If Eq. (2.19) holds, then for the basis operators $A_{jk}^{(i)}$ $(i = 1, 2)$ associated with the two processors we have

$$A_{jk}^{(2)} = \sum_{m,n=1}^{N} (U_{p1})_{jm} (U_{p2})_{nk} A_{mn}^{(1)}. \tag{2.20}$$

Therefore, we can regard two processors whose set of operators $A_{jk}^{(i)}$ are related by the above equation as equivalent.

If the processors 1 and 2 are equivalent, then they will implement the same set of superoperators, i.e., $\mathcal{C}_{G_1} = \mathcal{C}_{G_2}$. In order to see this, suppose that when the state $|\Xi_1\rangle_p$ is sent into the program register of the processor 1, the map T_{Ξ_1}, with program operators $\{A_j^{(1)}(\Xi_1)\}$, is performed on the data state. Now let us consider what happens when we send the state $|\Xi_2\rangle_p = U_{p2}^{-1} |\Xi_1\rangle_p$ into the input of the program register of the processor 2. This will produce the mapping T_{Ξ_2} on the data state of the processor 2. The relation between the program operators $A_j^{(1)}(\Xi_1)$ and $A_j^{(2)}(\Xi_2)$ is

$$A_j^{(2)}(\Xi_2) = \sum_{k=1}^{N} (U_{p1})_{jk} A_1^{(1)}(\Xi_1)\,. \tag{2.21}$$

The operators $A_j^{(1)}(\Xi_1)$ are Kraus operators for the mapping T_{Ξ_1} and the operators $A_j^{(2)}(\Xi_2)$ are Kraus operators for the mapping T_{Ξ_2}. The above equation implies that the mappings are identical, $T_{\Xi_1} = T_{\Xi_2}$.[10] Therefore, any superoperator that can be realized by the processor 1 can also be realized by the processor 2. Similarly, it can be shown that any superoperator that can be realized by the processor 2 can also be realized by processor the 1. This shows that the two processors implement the same set of superoperators.

Here we summarize some of our results so far:

- For a given processor, G, any member of the class of all possible completely positive linear maps realizable by G, $\mathcal{C}_G$, can be expressed in terms of the operators A_{jk}.
- We can mimic the action induced by any mixed program state by a pure program state in a larger program space.
- For any two mappings realized by the processor G and the pure state programs $|\Xi_1\rangle_p$ and $|\Xi_2\rangle_p$ the identity

$$\sum_k A_k^\dagger(\Xi_1) A_k(\Xi_2) = \langle \Xi_1 | \Xi_2 \rangle I_d, \tag{2.22}$$

holds. This follows directly from Eqs. (2.6) and (2.7).

2.2. Classes of Processors

In what follows we will examine several different kinds of deterministic quantum processors. These will serve to illustrate some of the general considerations in the previous sections.

2.2.1. U Processors

Let us suppose that the eigenvectors of the unitary operator, G, that describes the fixed array of gates are tensor products. In particular, suppose that we have a single orthonormal basis for $\mathcal{H}_p$, $\{|k\rangle_p | k = 1, \ldots N\}$ and a collection of orthonormal bases for $\mathcal{H}_d$, $\{|\phi_{mk}\rangle_d | k = 1, \ldots N, m = 1, \ldots M\}$, where M is the dimension of $\mathcal{H}_d$. For each value of k, the vectors $\{|\phi_{mk}\rangle_d | m = 1, \ldots M\}$ form an orthonormal basis for $\mathcal{H}_d$. We call a processor a *U processor* if the eigenvectors of G, $|\Phi_{mk}\rangle_{dp}$, are of the form

$$|\Phi_{mk}\rangle_{dp} = |\phi_{mk}\rangle_d \otimes |k\rangle_p. \tag{2.23}$$

In this case the operators A_{jk} are given by $A_{jk} = \delta_{jk} U_j$ where U_j is unitary (its eigenstates are just $\{|\phi_{mj}\rangle_d | m = 1, \ldots M\}$) . This is the type of processor that was studied by Chuang and Nielsen,[7] and we recall that the dimension of $\mathcal{H}_p$ is equal to the number of unitary operators that this type of processor can perform. The processor acts on the state $|\psi\rangle_d \otimes |j\rangle_p$ as

$$G(|\psi\rangle_d \otimes |j\rangle_p) = (U_j |\psi\rangle_d) \otimes |j\rangle_p, \tag{2.24}$$

where $|\psi\rangle_d$ is an arbitrary data state.

For a general pure program state $|\Xi\rangle_p = \sum_j \alpha_j |j\rangle_p$ the encoded mapping, or the superoperator, T_Ξ, is given by the expression $T_\Xi[\varrho_d] = \sum_j |\alpha_j|^2 U_j \varrho_d U_j^\dagger$. In the case of a mixed program state $\varrho_p = \sum_{jk} R_{jk} |j\rangle\langle k|$ the data state is transformed as $T_{\varrho_p}[\varrho_d] = \sum_j R_{jj} U_j \varrho_d U_j^\dagger$. Comparing these two cases we conclude that we can always mimic a mixed program state by a pure one, in particular, it is enough to set $\alpha_j = \sqrt{R_{jj}}$. Hence, for this type of processor we can consider only pure program states without any loss of generality.

Finally, we note that for all program states $|\Xi\rangle_p$

$$T_\Xi \left[\frac{1}{d} I_d \right] = \sum_j |\alpha_j|^2 U_j \frac{1}{d} I_d U_j^\dagger = \frac{1}{d} I_d. \tag{2.25}$$

This implies that each element of $\mathcal{C}_G$ is *unital*, i.e., it maps the identity operator into itself.

2.2.2. Y Processors

A second possibility is to consider a situation that is in some way the reverse of the one we just examined. We have a single orthonormal basis for $\mathcal{H}_d$, $\{|m\rangle_d | m = 1, \ldots M\}$, and a set of orthonormal bases for $\mathcal{H}_p$, $\{|\chi_{mk}\rangle_p | k = 1, \ldots N\}$, where the index $m = 1, \ldots M$ labels the bases and the index k labels the individual basis elements. We again assume that the eigenvectors of G, $|\Phi_{mk}\rangle_{dp}$ are tensor products, but now they are given by

$$|\Phi_{mk}\rangle_{dp} = |m\rangle_d \otimes |\chi_{mk}\rangle_p. \tag{2.26}$$

In this case the processor can be expressed as $G = \sum_m |m\rangle_d\langle m| \otimes U_m$, where U_m is unitary and has eigenvectors $\{|\chi_{mk}\rangle_p | k = 1, \ldots N\}$. We find the operators A_{jk} by first choosing a single orthonormal basis in $\mathcal{H}_p$, $\{|k\rangle_p\}$, and computing

$$A_{jk} = {}_p\langle j|G|k\rangle_p = \sum_m |m\rangle\langle m|\langle j|U_m|k\rangle$$
$$= \sum_m (U_m)_{jk}|m\rangle\langle m|. \tag{2.27}$$

The maps produced by Y processors are unital, as can be seen from

$$\sum_j A_{jk_1}A^\dagger_{jk_2} = \sum_j \sum_{ab} (U_m)_{jk_1}(U_n^\dagger)_{jk_2}|m\rangle\langle m|n\rangle\langle n|$$
$$= \sum_{ja} (U_m)_{jk_1}(U_m^\dagger)_{jk_2}|m\rangle\langle m|$$
$$= \delta_{k_1k_2}\sum_m |m\rangle\langle m| = \delta_{k_1k_2}I \tag{2.28}$$

The action of a Y processor is particularly simple if all of the operators U_m have some common eigenstates, and the program state is one of them. Suppose that $U_m|\Xi\rangle_p = e^{i\phi_m}|\Xi\rangle_p$, then

$$G\left(\sum_m c_m|m\rangle_d \otimes |\Xi\rangle_p\right) = \left(\sum_m c_m e^{i\phi_m}|m\rangle_d\right)\otimes|\Xi\rangle_p. \tag{2.29}$$

In summary, we can say that both the U and Y processors are controlled-U gates; in the U processor, the control system is the program and the target is the data, and in the Y processor, it is the target that is the program and the control that is the data.

2.2.3. U' Processors

Let us consider a simple modification of the U processor, which we shall call the U' processor. Suppose we have two different orthonormal bases of $\mathcal{H}_p$, $\{|k\rangle_p\}$ and $\{|\chi_k\rangle_p\}$. We define a U' processor to have a unitary operator, G, of the form

$$G = \sum_k U_k \otimes |k\rangle_p\langle\chi_k|. \tag{2.30}$$

This looks like a new kind of processor, but it is actually equivalent to a U processor. This can be seen immediately if we realize that there exists a unitary operator, U_p, acting on $\mathcal{H}_p$ such that $|\chi_k\rangle_p = U_p|k\rangle_p$. Therefore, we have that

$$G = \left(\sum_k U_k \otimes |k\rangle_p\langle k|\right)(I_d \otimes U_p^\dagger), \tag{2.31}$$

so that G is, in fact, equivalent to a U processor.

2.2.4. Y' Processors

Now let us try a modification of the Y processor in the same spirit as the one we just made to the U processor. Suppose we have two different orthonormal bases of $\mathcal{H}_d$, $\{|m\rangle_p\}$ and $\{|\phi_m\rangle_d\}$. We define a Y' processor to have a unitary operator, G, of the form

$$G = \sum_m |m\rangle_d\langle\phi_m| \otimes U_m. \tag{2.32}$$

For the operators A_{jk} we obtain

$$A_{jk} = {}_p\langle j|G|k\rangle_p = \sum_m |m\rangle\langle\phi_m|(U_m)_{jk}. \tag{2.33}$$

This type of processor is not equivalent to a Y processor. It does, however, share the property of producing unital maps as can be seen from

$$\begin{aligned}\sum_j A_{jk_1}A^\dagger_{jk_2} &= \sum_{j,m,n} |m\rangle\langle\phi_m|\phi_n\rangle\langle n|(U_m)_{jk_1}(U_n^\dagger)_{k_2 j} \\ &= \sum_{j,m} |m\rangle\langle m|(U_m^\dagger)_{k_2 j}(U_m)_{jk_1} \\ &= \delta_{k_1k_2}\sum_m |m\rangle\langle m| \\ &= \delta_{k_1k_2}I_d,\end{aligned} \tag{2.34}$$

which implies that for any program state, the identity on $\mathcal{H}_d$ is mapped into itself.

2.2.5. Covariant Processors

Another class of processors that may be of interest are *covariant* processors. Covariance has proven to be an important property in the study of quantum machines. Covariant processors have the property that if the processor maps the input data state $\varrho_{\text{in}} = |\psi\rangle_d\,{}_d\langle\psi|$, which we shall assume is a qudit, onto the output density matrix ρ_{out}, then it maps the input state $U|\psi\rangle_d$ onto the output density matrix $U\rho_{\text{out}}U^{-1}$, for all $U \in \mathcal{G}$, where $\mathcal{G}$ is a subgroup of $SU(D)$, for some subset $\mathcal{S}$ of all possible program states.[7] This relation implies that if $|\Xi\rangle \in \mathcal{S}$, then the operators $A_j(\Xi)$ satisfy the relation

[7] Whether there are any non-trivial covariant processors in the case that $\mathcal{S} = \mathcal{H}_p$ is an open question.

$$\sum_{j=1}^{N} U A_j(\Xi)\varrho_{\text{in}} A_j^{\dagger}(\Xi)U^{-1} = \sum_{j=1}^{N} A_j(\Xi)U\varrho_{\text{in}}U^{-1}A_j^{\dagger}(\Xi), \tag{2.35}$$

for all $U \in \mathcal{G}$. Let us now consider the case $\mathcal{G} = SU(D)$. If we take ρ_{in} to be I_d/d, we find

$$\sum_{j=1}^{N} U A_j(\Xi)A_j^{\dagger}(\Xi)U^{-1} = \sum_{j=1}^{N} A_j(\Xi)A_j^{\dagger}(\Xi). \tag{2.36}$$

Because this holds for all $U \in SU(D)$, Schur's Lemma implies that

$$\sum_{j=1}^{N} A_j(\Xi)A_j^{\dagger}(\Xi) = c\, I, \tag{2.37}$$

where c is a constant. Taking the trace of both sides of Eq. (2.37) we find

$$\text{Tr}\left(\sum_{j=1}^{N} A_j(\Xi)A_j^{\dagger}(\Xi)\right) N = c\,\text{Tr}(I) = c\,N, \tag{2.38}$$

so that $c = 1$. Because this relation holds for any program state, we have that

$$\sum_{j=1}^{N} A_{jk_1}A_{jk_2}^{\dagger} = \delta_{k_1k_2} I_d, \tag{2.39}$$

which implies that the maps produced by a processor that is covariant with respect to $SU(D)$ are unital. As an example of covariant programmable quantum processor one can consider the so called quantum information distributor as described in Sec. 3.3.

3. PROBABILISTIC PROGRAMMABLE PROCESSORS

As shown by Nielsen and Chuang[7] universal deterministic programmable processors of finite extent do not exist. On the other hand, universal probabilistic programmable processors can be designed. In this section we will address a probabilistic implementation of an operation U, encoded in the state of a program register $|\Xi_U\rangle_p$, on the data state $|\psi\rangle_d$. The probabilistic character of the implementation of the processor is related to the

fact that the program register is measured at the output of the processor—see Fig. 2. We will present a simple example of how to apply an arbitrary operation to a single qubit initially prepared in a state $|\psi\rangle$. The gate array consists of four C-NOT gates, and can implement four programs perfectly. These programs cause the one of the operations I, σ_x, $-i\sigma_y$, or σ_z to be performed on the data qubit. Here I is the identity and σ_j, where $j = x, y, z$ is a Pauli matrix. By choosing programs that are linear combinations of the four basic ones, it is possible to probabilistically perform any linear operation on the data qubit. We generalize the idea to an arbitrary dimensional quantum system, a qudit.

3.1. Operations on Qubits

We would like to construct a device that will do the following: The input consists of a qubit, $|\psi\rangle_d$, and a second state, $|\Xi_U\rangle_p$, which may be a multiqubit state, that acts as a program. The output of the device will be a state $U|\psi\rangle_d$, where U is an operation that is specified by $|\Xi_U\rangle_p$. In order to make this a little less abstract, we first consider an example: Let $|\phi\rangle$ and $|\phi_\perp\rangle$ be two orthogonal qubit states, and suppose that we want to perform the operation

$$A_z = |\phi_\perp\rangle\langle\phi_\perp| - |\phi\rangle\langle\phi| = I - 2|\phi\rangle\langle\phi|, \tag{3.1}$$

on $|\psi\rangle_d$. The action of this operator is analogous to that of σ_z in the basis $\{|0\rangle, |1\rangle\}$, except that it acts in the basis $\{|\phi_\perp\rangle, |\phi\rangle\}$. That is, σ_z does nothing to $|0\rangle$ and multiplies $|1\rangle$ by -1, while A_z does nothing to $|\phi_\perp\rangle$ and multiplies $|\phi\rangle$ by -1. Can we find a network and a program vector to implement this operation on $|\psi\rangle_d$?

We can, in fact, do this by using the network for a QID as introduced in Ref. 24 (this is a modification of the quantum cloning transformation[25,26] see more details in Sec. 3.3). In this network the program register is represented by a two qubit state $|\Xi_A\rangle_p$. Before we present the network for the programmable gate array, we shall introduce notation for its components. A C-NOT gate D_{jk} acting on qubits j and k performs the transformation,

$$D_{jk}|m\rangle_j|n\rangle_k = |m\rangle_j|m \oplus n\rangle_k, \tag{3.2}$$

where j is the control bit, k is the target bit, and m and n are either 0 or 1. The addition is modulo 2. The QID network consists of four C-NOT gates, and acts on three qubits (a single data qubit denoted by a subscript 1 and two program qubits denoted by subscripts 2 and 3, respectively). Its action is given by the operator $P_{123} = D_{31}D_{21}D_{13}D_{12}$ (for more details see

Sec. 3.3). As our first task, we shall determine how this network acts on input states where qubit 1 is in the state $|\psi\rangle$, and qubits 2 and 3 are in Bell basis states. The Bell basis states are defined by

$$\begin{aligned}
|\Phi_+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \equiv |\Xi_{01}\rangle\,, \\
|\Phi_-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \equiv |\Xi_{11}\rangle\,; \\
|\Psi_+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \equiv |\Xi_{00}\rangle\,; \\
|\Psi_-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \equiv |\Xi_{10}\rangle.
\end{aligned} \tag{3.3}$$

We find that

$$\begin{aligned}
P_{123}|\psi\rangle_1|\Phi_+\rangle_{23} &= (\sigma_x|\psi\rangle_1)|\Phi_+\rangle; \\
P_{123}|\psi\rangle_1|\Phi_-\rangle_{23} &= (-i\sigma_y|\psi\rangle_1)|\Phi_-\rangle; \\
P_{123}|\psi\rangle_1|\Psi_+\rangle_{23} &= |\psi\rangle_1|\Psi_+\rangle\,; \\
P_{123}|\psi\rangle_1|\Psi_-\rangle_{23} &= (\sigma_z|\psi\rangle_1)|\Psi_-\rangle.
\end{aligned} \tag{3.4}$$

Any operation on qubits can be expanded in terms of Pauli matrixes and the identity. The above equations mean that the Bell basis vectors are "programs" for a complete set of operations. In order to see how to make use of this, let us expand our proposed operation in terms of this complete set. Expressing $|\phi\rangle$ as $|\phi\rangle = \mu|0\rangle + \nu|1\rangle$, we have that

$$\begin{aligned}
A_z = I - 2|\phi\rangle\langle\phi| &= \begin{pmatrix} |\nu|^2 - |\mu|^2 & -2\mu\nu^* \\ -2\mu^*\nu & |\mu|^2 - |\nu|^2 \end{pmatrix}, \\
&= -(\mu\nu^* + \mu^*\nu)\sigma_x + (\mu\nu^* - \mu^*\nu)(-i\sigma_y) \\
&\quad + (|\nu|^2 - |\mu|^2)\sigma_z.
\end{aligned} \tag{3.5}$$

We can now apply the operation A to $|\psi\rangle$ by sending in the "program" vector

$$\begin{aligned}
|\Xi_A\rangle_{23} = &- (\mu\nu^* + \mu^*\nu)|\Phi_+\rangle_{23} + (\mu\nu^* - \mu^*\nu)|\Phi_-\rangle_{23} \\
&+ (|\nu|^2 - |\mu|^2)|\Psi_-\rangle_{23},
\end{aligned} \tag{3.6}$$

and measuring the program outputs in order to determine if they are in the state $(|\Phi_+\rangle + |\Phi_-\rangle + |\Psi_-\rangle)/\sqrt{3}$. If they are, our operation has been accomplished. Note that the measurement is independent of the vector $|\phi\rangle$ so that no knowledge of this vector is necessary to make the measurement

and to determine whether the procedure has been successful. As we see, the probability of success is 1/3 for the implementation of the operation A_z which is parameterized in general by two continuous parameters (i.e., the state $|\phi\rangle$).

Let us examine the program vector more carefully. If we define the unitary operation, U_{init}, by

$$\begin{aligned} U_{\text{init}}|00\rangle &= -|10\rangle; \\ U_{\text{init}}|10\rangle &= -|11\rangle; \\ U_{\text{init}}|11\rangle &= |01\rangle; \\ U_{\text{init}}|01\rangle &= |00\rangle; \end{aligned} \tag{3.7}$$

we have that

$$|\Xi_A\rangle_{12} = U_{\text{init}} \frac{1}{\sqrt{2}}(|\phi\rangle|\phi_\perp\rangle + |\phi_\perp\rangle|\phi\rangle). \tag{3.8}$$

Finally, we can summarize our procedure. The steps are

1. Start with the state $\frac{1}{\sqrt{2}}(|\phi\rangle|\phi_\perp\rangle + |\phi_\perp\rangle|\phi\rangle)$.
2. Apply U_{init}.
3. Send the resulting state into the control ports (inputs 2 and 3) and $|\psi\rangle$ into port 1.
4. Measure $(|\Phi_+\rangle + |\Phi_-\rangle + |\Psi_-\rangle)/\sqrt{3}$ at the output of the control ports.
5. If the result is yes, then the output of port 1 is $(I - 2|\phi\rangle\langle\phi|)|\psi\rangle$.

Before proceeding to a more general consideration of this network, let us make an observation. Suppose that we carry out the same procedure, but instead of starting with the program vector $(|\phi\rangle|\phi_\perp\rangle + |\phi_\perp\rangle|\phi\rangle)/\sqrt{2}$, we start instead with the program vector $(|\phi\rangle|\phi\rangle - |\phi_\perp\rangle|\phi_\perp\rangle)/\sqrt{2}$. At the end of the procedure the output of the data register is $A_x|\psi\rangle$, where

$$A_x = |\phi\rangle\langle\phi_\perp| + |\phi_\perp\rangle\langle\phi|. \tag{3.9}$$

The operation A_x interchanges $|\phi\rangle$ and $|\phi_\perp\rangle$. Its action is analogous to that of σ_x, which interchanges the vectors $|0\rangle$ and $|1\rangle$. The probability of success for this procedure is also 1/3.

We now need to determine whether there is a program for any operator that could act on $|\psi\rangle$. The operator need not be unitary; it could be a result of coupling $|\psi\rangle$ to an ancilla, evolving the coupled system (a unitary process), and then measuring the ancilla. Therefore, if A is now any

linear operator acting on a two-dimensional quantum system, the transformations in which we are interested are given by

$$|\psi\rangle \to \frac{1}{\|A\psi\|} A|\psi\rangle. \tag{3.10}$$

Let us denote the operators, which can be implemented by Bell state programs, by $S_{00}=I$, $S_{01}=\sigma_x$, $S_{10}=\sigma_z$, and $S_{11}=-i\sigma_y$. Any 2×2 matrix can be expanded in terms of these operators, so that we have

$$A=\sum_{j,k=0}^{1} \tilde{a}_{jk} S_{jk}. \tag{3.11}$$

We now define $a_{jk}=\tilde{a}_{jk}/\sqrt{\eta}$, where

$$\eta=\sum_{j,k=0}^{1} |\tilde{a}_{jk}|^2, \tag{3.12}$$

so that

$$1=\sum_{j,k=0}^{1} |a_{jk}|^2. \tag{3.13}$$

Now let us go back to our network and consider the program vector given by

$$|\Xi_A\rangle=\sum_{j,k=0}^{1} a_{jk}|\Xi_{jk}\rangle, \tag{3.14}$$

and at the output of the program register we shall measure the projection operator corresponding to the vector $(1/2)\sum_{j,k=0}^{1}|\Xi_{jk}\rangle$. If the measurement is successful, the state of the data register is, up to normalization, given by

$$|\psi\rangle \to \left(\sum_{j,k=0}^{1} a_{jk} S_{jk}\right)|\psi\rangle\,. \tag{3.15}$$

After this state is normalized, it is just $(1/\|A\psi\|)|\psi\rangle$. This means that for any transformation of the type given in Eq. (3.10) satisfying the normalization $||A||=(1/N)\mathrm{Tr}A^{\dagger}A=1$, we can find a program for our network that will carry it out.

3.2. Generalization to Qudits

In order to extend the network presented above to higher dimensions, we must first introduce a generalization of the two-qubit C-NOT gate[24] (see also Ref. 27). To make our discussion self-contained we first present a brief review of the formalism describing quantum states in a finite-dimensional Hilbert space. Here we follow the notation introduced in Ref. 28. Let the N-dimensional Hilbert space be spanned by N orthogonal normalized vectors $|x_k\rangle$ or, equivalently, by N vectors $|p_l\rangle$, $k, l = 0, \ldots, N-1$, where these bases are related by the discrete Fourier transform

$$\begin{aligned} |x_k\rangle &= \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \exp\left(-i \frac{2\pi}{N} kl\right) |p_l\rangle ; \\ |p_l\rangle &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(i \frac{2\pi}{N} kl\right) |x_k\rangle . \end{aligned} \tag{3.16}$$

Without loss of generality, it can be assumed that these bases consist of sets of eigenvectors of non-commuting operators X and P:

$$X|x_k\rangle = k|x_k\rangle , \quad P|p_l\rangle = l|p_l\rangle , \tag{3.17}$$

that is,

$$X = \sum_{k=0}^{N-1} k|x_k\rangle\langle x_k| ; \quad P = \sum_{l=0}^{N-1} l|p_l\rangle\langle p_l| . \tag{3.18}$$

For instance, we can assume that the operators X and P are related to a discrete "position" and "momentum" of a particle on a ring with a finite number of equidistant sites.[29] Specifically, we can introduce a length scale, L, and two operators, the position x and the momentum p, such that

$$x|x_k\rangle = x_k|x_k\rangle , \quad p|p_l\rangle = p_l|p_l\rangle , \tag{3.19}$$

where

$$x_k = L\sqrt{\frac{2\pi}{N}} k; \quad p_l = \frac{1}{L}\sqrt{\frac{2\pi}{N}} l , \tag{3.20}$$

where we have used units such that $\hbar = 1$. The length, L can, for example, be taken equal to $\sqrt{1/\omega m}$, where m is the mass and ω the frequency of a quantum "harmonic"oscillator within a finite dimensional Fock space.

The squared absolute values of the scalar product of eigenkets (3.17) do not depend on the indices k, l:

$$|\langle x_k | p_l \rangle|^2 = 1/N\,, \tag{3.21}$$

which means that pairs (k,l) form a discrete phase space (i.e., pairs (k,l) represent "points" of the discrete phase space) on which (quasi)-probability density distributions associated with a given quantum state can be defined.[30–34] Next we introduce operators which shift (cyclicly permute) the basis vectors[35]:

$$\begin{aligned} R_x(n)|x_k\rangle &= |x_{(k+n) \bmod N}\rangle; \\ R_p(m)|p_l\rangle &= |p_{(l+m) \bmod N}\rangle, \end{aligned} \tag{3.22}$$

where the sums of indices are taken modulo N (this summation rule is considered throughout this paper, where it is clear we will not explicitly write the symbol $\bmod N$). For more about the properties of these operators and the role they play in the discrete phase space (k,l) see Ref. 36.

A general single-particle state in the x-basis can be expressed as

$$|\Psi\rangle_1 = \sum_{k=0}^{N-1} c_k |x_k\rangle_1\,; \qquad \sum_{k=0}^{N-1} |c_k|^2 = 1\,. \tag{3.23}$$

The basis of maximally entangled two-particle states (the analogue of the Bell basis for spin-$\frac{1}{2}$ particles) can be written as

$$|\Xi_{mn}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(i \frac{2\pi}{N} mk\right) |x_k\rangle |x_{(k-n) \bmod N}\rangle\,, \tag{3.24}$$

where $m, n = 0, \ldots, N-1$. We can also rewrite these maximally entangled states in the p-basis:

$$|\Xi_{mn}\rangle = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \exp\left(-i \frac{2\pi}{N} nl\right) |p_{(m-l) \bmod N}\rangle |p_l\rangle\,. \tag{3.25}$$

The states $|\Xi_{mn}\rangle$ form an orthonormal basis

$$\langle \Xi_{kl} | \Xi_{mn} \rangle = \delta_{k,m} \delta_{l,n}\,, \tag{3.26}$$

with

$$\sum_{m,n=0}^{N-1} |\Xi_{mn}\rangle\langle\Xi_{mn}| = I \otimes I\,. \tag{3.27}$$

In order to prove the above relations we have used the standard relation $\sum_{n=0}^{N-1} \exp[2\pi i(k-k')n/N] = N\delta_{k,k'}$.

It is interesting to note that the whole set of N^2 maximally entangled states $|\Xi_{mn}\rangle$ can be generated from the state $|\Xi_{00}\rangle_{23}$ by the action of *local* unitary operations (shifts), e.g.,

$$|\Xi_{mn}\rangle_{23} = I_2 \otimes R_x^{\dagger}(n) R_p(m) |\Xi_{00}\rangle_{23}\,, \tag{3.28}$$

acting just on system 3 in this particular case.

From the definition of the states $|\Xi_{mn}\rangle_{23}$ it follows that they are simultaneously eigenstates of the operators $X_2 - X_3$ and $P_2 + P_3$:

$$\begin{aligned} (X_2 - X_3)|\Xi_{mn}\rangle_{23} &= n|\Xi_{mn}\rangle_{23}; \\ (P_2 + P_3)|\Xi_{mn}\rangle_{23} &= m|\Xi_{mn}\rangle_{23}. \end{aligned} \tag{3.29}$$

We easily see that for $N = 2$ the above formalism reduces to the well-known spin-$\frac{1}{2}$ particle (qubit) case.

Now we introduce generalizations of the two-qubit C-NOT gate (see also Ref. 37). In the case of qubits the C-NOT gate is represented by a two-particle operator such that if the first (control) particle labeled a is in the state $|0\rangle$ nothing "happens" to the state of the second (target) particle labeled b. If, however, the control particle is in the state $|1\rangle$ then the state of the target is "flipped", i.e., the state $|0\rangle$ is changed into the state $|1\rangle$ and vice versa. Formally we can express the action of this C-NOT gate as a two-qubit operator of the form

$$D_{ab} = \sum_{k,m=0}^{1} |k\rangle_a\langle k| \otimes |(m+k) \bmod 2\rangle_b\langle m|\,. \tag{3.30}$$

We note that in principle one can introduce an operator $D_{ab}^{\dagger}$ defined as

$$D_{ab}^{\dagger} = \sum_{k,m=0}^{1} |k\rangle_a\langle k| \otimes |(m-k) \bmod 2\rangle_b\langle m|\,. \tag{3.31}$$

In the case of qubits these two operators are equal. This is not the case when the dimension of the Hilbert space is larger than 2.[37] Let us generalize the above definition of the operator D for $N > 2$. Before doing so, we

shall simplify our notation. Because we will work mostly in the x-basis we shall use the notation $|x_k\rangle \equiv |k\rangle$ where it may be done so unambiguously. With this in mind we now write

$$D_{ab} = \sum_{k,m=0}^{N-1} |k\rangle_a\langle k| \otimes |(m+k) \bmod N\rangle_b\langle m| \,. \tag{3.32}$$

From the definition (3.32) it follows that the operator D_{ab} acts on the basis vectors as (see Fig. 6)

$$D_{ab}|k\rangle|m\rangle = |k\rangle|(k+m) \bmod N\rangle \,, \tag{3.33}$$

which means that this operator is equal to the conditional adder[38, 39] and can be performed with the help of a simple quantum network as discussed in Ref. 38.

If we take into account the definition of the shift operator $R_x(n)$ given by Eq. (3.22) and the definition of the position and momentum operators x and p given by Eq. (3.19) we can rewrite the operator D_{ab} as:

$$\begin{aligned} D_{ab} &= \sum_{k,m=0}^{N-1} |k\rangle_a\langle k| \otimes R_x^{(b)}(k)|m\rangle_b\langle m| \\ &\equiv \sum_{k=0}^{N-1} |k\rangle_a\langle k| \otimes R_x^{(b)}(k) \,, \end{aligned} \tag{3.34}$$

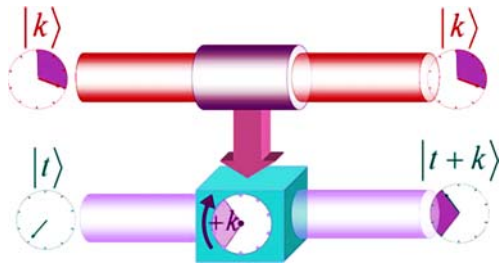

Fig. 6. A schematic description of the two-qudit conditional-shift gate. The arrow between two qudit lines indicates the action of the control: The control qudit that is prepared in the state $|k\rangle$ acts on the target qudit that is prepared in the state $|t\rangle$.

and analogously

$$D_{ab}^{\dagger} = \sum_{k,m=0}^{N-1} |k\rangle_a\langle k| \otimes |(m-k) \bmod N\rangle_b\langle m| \equiv \sum_{k=0}^{N-1} |k\rangle_a\langle k| \otimes R_x^{(b)}(-k)\,, \tag{3.35}$$

where the subscripts a and b indicate on which Hilbert space the given operator acts. Now we see that for $N > 2$ the two operators D and $D^{\dagger}$ do differ; they describe conditional shifts in opposite directions. We see that the generalization of the C-NOT operator are the *conditional shifts*. The amount by which the target (in our case particle b) is shifted depends on the state of the control particle (a) [for a pictorial representation of this gate see Fig. 6].

3.3. Quantum Information Distributor

As shown in Ref. 24 quantum control over the quantum information can be achieved with the help of a quantum "machine," the so-called QID. The machine takes as an input a system qudit prepared in an unknown state $|\Psi\rangle_1$ and two ancilla qudits prepared in the state $|\Theta\rangle_{23}$ that plays the role of quantum program (i.e., the CP map that has to be performed on the system qubit is encoded in this state). The action of the QID itself is described by a unitary operator P_{123} acting on the Hilbert space that is a tensor product of the three qudits under consideration. This unitary operator can be expressed as a sequence of four controlled shifts D_{kl}, i.e.,

$$P_{123} = D_{31} D_{21}^{\dagger} D_{13} D_{12}\,. \tag{3.36}$$

The flow of information in the quantum distributor, as described by the unitary operator (3.36), is governed by the preparation of the distributor itself, i.e., by the choice of the program state $|\Theta\rangle_{23}$. In other words, we imagine the transformation (3.36) as a universal "processor" or distributor and the state $|\Theta\rangle_{23}$ as "program" through which the information flow is controlled.

We present the logical network for the QID in Fig. 7. The output state of the three particle system after the four controlled shifts are applied is

$$|\Omega^{(\mathrm{out})}\rangle_{123} = D_{31} D_{21}^{\dagger} D_{13} D_{12} |\Psi\rangle_1 |\Theta\rangle_{23}\,. \tag{3.37}$$

Note that the QID is covariant with respect to any choice of the state $|\Psi\rangle_1$ of data register (for more details see Ref. 24).

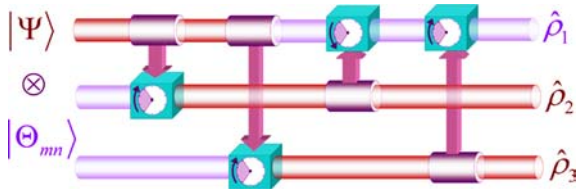

Fig. 7. A logic network for the universal quantum processor as given by the unitary transformation (3.36). The action of the controlled shift operator D_{jk} is represented in Fig. 6.

3.3.1. *Factorized Program States*

Let us first assume that the two program qudits are in a pure state

$$|\Theta\rangle_{23} = |x_m\rangle_2 |p_n\rangle_3 \,. \tag{3.38}$$

After the action of the QID the state $|\Omega\rangle_{123} = |\Psi\rangle_1 \otimes |\Theta\rangle_{23}$ transforms as

$$\begin{aligned} |\Omega^{(\mathrm{out})}\rangle_{123} &= P_{123}|\Psi\rangle_1 |x_m\rangle_2 |p_n\rangle_3 \\ &= \left[R_x(m)R_p^{\dagger}(n)|\Psi\rangle_2\right] \otimes |\Xi_{nm}\rangle_{31}. \end{aligned} \tag{3.39}$$

So we can observe two actions of the QID on the input state: Firstly, the state of the original qudit has been totally copied on the state of the second qudit (this might be considered as the swap operation). Simultaneously, the second qudit undergoes two rotations described by the operator $R_x(m)R_p^{\dagger}(n)$, where the values of the rotations are uniquely determined by the program state. Finally, the two remaining qudits (labeled as 1 and 3) became maximally entangled as the result of the action of the QID.

3.3.2. *Maximally Entangled Program States*

Let us assume that the QID state $|\Theta\rangle_{23}$ is initially prepared in the maximally entangled state $|\Xi_{mn}\rangle_{23}$ given by Eq. (3.24) Taking the original system to be prepared in the state $|\Psi\rangle_1$, i.e., the three qudits at the input are in the state

$$|\Omega\rangle_{123} = |\Psi\rangle_1 \otimes |\Xi_{mn}\rangle_{23} \tag{3.40}$$

we find after the QID transformation the expression for the state vector of the three qudits

$$|\Omega^{(\mathrm{out})}\rangle_{123} = \left[R_x^{\dagger}(n)R_p^{\dagger}(m)|\Psi\rangle_1\right] \otimes |\Xi_{mn}\rangle_{23} \,. \tag{3.41}$$

We see that if the program register is initially prepared in the maximally entangled state then the information encoded in the input state of the first (system) qudit will remain in this qudit, but the QID will induce a specific rotation on this qudit that is uniquely determined by the maximally entangled state of the program qudits. Interestingly enough, the program state is not changed at all in this case.

3.4. QID as Universal Processor

We assume the network for the probabilistic universal quantum processor to be the QID as described by Eq. (3.36) as a sequence of four conditional shifts gates D. The sequence of four operators acting on the basis vectors gives $|n\rangle_1|m\rangle_2|k\rangle_3$ as

$$\begin{aligned} &D_{31}D_{21}^{\dagger}D_{13}D_{12}|n\rangle_1|m\rangle_2|k\rangle_3 \\ &\quad = |(n-m+k)\bmod N\rangle_1\,|(m+n)\bmod N\rangle_2\,|(k+n)\bmod N\rangle_3\,. \end{aligned} \tag{3.42}$$

We now turn to the fundamental program states. A basis consisting of maximally entangled two-particle states (the analogue of the Bell basis for spin-$\frac{1}{2}$ particles) is given by[40]

$$|\Xi_{mn}\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\exp\Big(i\frac{2\pi}{N}mk\Big)|k\rangle|(k-n)\bmod N\rangle, \tag{3.43}$$

where $m,n=0,\ldots,N-1$. If $|\Xi_{mn}\rangle_p$ is the initial state of the program register, and $|\Psi\rangle=\sum_j \alpha_j|j\rangle_d$ (here, as usual, $\sum_j|\alpha_j|^2=1$) is the initial state of the data register, it then follows that

$$\begin{aligned} &P_{123}|\Psi\rangle_1|\Xi_{mn}\rangle_{23} \\ &\quad = \sum_{jk}\frac{\alpha_j}{\sqrt{N}}\exp\frac{2\pi ikm}{N}P_{123}|j\rangle|k\rangle|k-n\rangle \\ &\quad = \sum_{jk}\frac{\alpha_j}{\sqrt{N}}\exp\frac{2\pi ikm}{N}|j-n\rangle|k+j\rangle|k+j-n\rangle \\ &\quad = \sum_{jk}\alpha_j\exp\frac{-2\pi ijm}{N}|j-n\rangle|\Xi_{mn}\rangle \\ &\quad = (U^{(mn)}|\Psi\rangle)|\Xi_{mn}\rangle, \end{aligned} \tag{3.44}$$

where we have introduced the notation

$$U^{(mn)} = \sum_{s=0}^{N-1}\exp\frac{-2i\pi sm}{N}|s-n\rangle\langle s|. \tag{3.45}$$

This result is similar to the one we found in the case of a single qubit (see Sec. 3.1). We would now like to examine which transformations we can perform on the state in the data register by using a program consisting of a linear combination of the vectors $|\Xi_{mn}\rangle$ followed by the action of the processor P_{123} and a subsequent measurement of the program register.

The operators $U^{(mn)}$ satisfy the orthogonality relation

$$\mathrm{Tr}\left[(U^{(m'n')})^{\dagger}U^{(mn)}\right] = N\delta_{mm'}\delta_{nn'}. \tag{3.46}$$

The space of linear operators $\mathcal{T}(\mathcal{H})$ defined on some Hilbert space $\mathcal{H}$ with the scalar product given by (3.46) we know as *Hilbert–Schmidt space*. Thus the unitary operators $U^{(mn)}$ form an orthogonal basis in it and any operator $A \in \mathcal{T}(\mathcal{H})$ can be expressed in terms of them

$$A = \sum_{m,n=0}^{N-1} q_{mn}U^{(mn)}. \tag{3.47}$$

The orthogonality relation allows us to find the expansion coefficients in terms of the operators

$$q_{mn} = \frac{1}{N}\mathrm{Tr}\left[\left(U^{(mn)}\right)^{\dagger}A\right]. \tag{3.48}$$

Equations (3.46) and (3.47) imply that

$$\sum_{m,n=0}^{N-1} |q_{mn}|^2 = \frac{1}{N}\mathrm{Tr}(A^{\dagger}A). \tag{3.49}$$

Therefore, the program vector that implements the operator A is given by

$$|v_A\rangle_{23} = \left[\frac{N}{\mathrm{Tr}(A^{\dagger}A)}\right]^{1/2} \sum_{m,n=0}^{N-1} q_{mn}|\Xi_{mn}\rangle_{23}. \tag{3.50}$$

Application of the processor to the input state $|\Psi\rangle_1|v_A\rangle_{23}$ yields the output state

$$|\Omega\rangle_{123} = \sum_{mn} q_{mn}U^{(mn)}|\Psi\rangle_1 \otimes |\Xi_{mn}\rangle_{23}. \tag{3.51}$$

To obtain the final result we perform a projective measurement of the program register onto vector $|M\rangle_{23}$

$$|M\rangle = \frac{1}{N}\sum_{m,n=0}^{N-1} |\Xi_{mn}\rangle \tag{3.52}$$

If the outcome of the measurement is positive, then we get the required transformation A acting on an unknown, arbitrary input state $|\Psi\rangle_1$.

Let us consider an example. Suppose we choose for A the unitary operator $I-2|\phi\rangle\langle\phi|$, where the normalized state $|\phi\rangle$ can be expressed as

$$|\phi\rangle = \sum_{k=0}^{N-1} \beta_k |k\rangle. \tag{3.53}$$

The expansion coefficients for this operation are given by

$$q_{mn} = \delta_{m0}\delta_{n0} - \frac{2}{N}\sum_{k=0}^{N-1} e^{2\pi ikm/N}\beta_k^*\beta_{k-n}, \tag{3.54}$$

and the program vector for this operation is

$$|\Phi\rangle_{23} = |\Xi_{00}\rangle_{23} - \frac{2}{\sqrt{N}}\sum_{k,n=0}^{N-1} \beta_{-k}^*\beta_{-(k+n)}|k\rangle_2|k-n\rangle_3. \tag{3.55}$$

The program vector can be obtained from a state more closely related to $|\phi\rangle$ if we introduce a new unitary operator and a "complex conjugate" vector. Define the operator W by

$$W|k\rangle = |-k\rangle, \tag{3.56}$$

and the vector $|\phi^*\rangle$ by

$$|\phi^*\rangle = \sum_{k=0}^{N-1} \beta_k^*|k\rangle. \tag{3.57}$$

We then have that

$$|\Phi\rangle_{23} = (W_2\otimes I_3)\left(D_{23}^\dagger\right)^2\left(|\Xi_{00}\rangle_{23} - \frac{2}{\sqrt{N}}|\phi^*\rangle_2|\phi\rangle_3\right). \tag{3.58}$$

A network that performs the operation $(W_2\otimes I_3)\left(D_{23}^\dagger\right)^2$ could be added to the input of the program register so that the simpler state that appears on the right-hand side of Eq. (3.58) could be used as the program. At the output of the processor we have to perform the projective measurement discussed in the previous paragraph, and the probability of achieving the desired result is the same as the probability of successfully implementing the transformation, A. In this case the probability is $1/N^2$.

3.5. Success Probability

The probability, p, of successfully applying the operator A to the state $|\Psi\rangle_1$ in our example is rather small. This is because the operator we chose was a linear combination of all of the operators $U^{(mn)}$. This means that if the data register consists of l qubits, i.e., $N=2^l$, then the probability of a successful implementation of a general transformation A decreases exponentially with the size of the data register. However, if we were to choose an operator, or set of operators, that was a linear combination of only a few of the $U^{(mn)}$, then the success probability can be significantly improved. This would entail making a different measurement at the output of the program register. Instead of making a projective measurement onto the vector $|M\rangle$, one would instead make a measurement onto the vector

$$|M'\rangle = \frac{1}{\mathcal{N}^{1/2}} \sum_{m,n:q_{mn}\neq 0} |\Xi_{mn}\rangle \tag{3.59}$$

where $\mathcal{N}$ is the total number of nonzero coefficients q_{mn}, in the decomposition in Eq. (3.47). If the operation being implemented is unitary, then, in this case, the probability of implementing it is

$$p = \frac{1}{\mathcal{N}}, \tag{3.60}$$

where $\mathcal{N}$ is the total number of nonzero coefficients q_{mn}, in the decomposition (3.47). There are, in fact, large classes of operations that can be expressed in terms of a small number of operators $U^{(mn)}$.[8] For these operators, the probability of success can be relatively large and, in principle, independent of the size of the Hilbert space of the data register.

Example 1.

Let us consider the one-parameter set of unitary transformations U_φ

$$U_\varphi = \cos\varphi\, I + i \sin\varphi \left[\frac{1+i}{2} U^{(01)} + \frac{1-i}{2} U^{(03)} \right], \tag{3.61}$$

where the unitaries $U^{(mn)}$ are given by Eq. (3.45). These unitaries for $N=4$ can be explicitly written as

$$U^{(01)} = \sum_{s=0}^{3} (-i)^s P_s\,; \quad U^{(03)} = \sum_{s=0}^{3} (i)^s P_s\,, \tag{3.62}$$

[8] We note that an arbitrary sum of unitary operators $U^{(mn)}$ is not necessarily a unitary operator.

where $P_s = |s\rangle\langle s|$. From here we find the expression for the operator (3.61) in the form:

$$U_\varphi = \cos\varphi\, I + i \sin\varphi\, [P_0 + P_1 - P_2 - P_3]\,, \tag{3.63}$$

We note that if we rewrite the parameters s as binary numbers, $s = j_1 2 + j_0$, where j_k is either 0 or 1, and express the states $|s\rangle$ as tensor products of qubits, i.e., $|s\rangle = |j_1\rangle \otimes |j_0\rangle$, we find that the operator in brackets on the right-hand side of Eq. (3.61) can be expressed as

$$\left[\frac{1+i}{2} U^{(01)} + \frac{1-i}{2} U^{(03)}\right] = \sigma_3 \otimes I. \tag{3.64}$$

From Eq. (3.63) it is clear that U_φ has eigenvalues of magnitude 1, which implies that U_φ is unitary. It can be realized by the universal quantum processor (3.39) with a probability of successful implementation equal to 1/3. This example illustrates that it is possible to realize large classes of unitary operations with a probability that is greater than the reciprocal of the dimension of the program register.

This example can be easily generalized. Consider a one-parameter set of unitary operators acting on a Hilbert space consisting of l qubits, which is given by

$$U_\varphi = \cos\varphi\, I^{\otimes l} + i \sin\varphi\, \sigma_3 \otimes I^{\otimes(l-1)}. \tag{3.65}$$

The operator $\sigma_3 \otimes I^{\otimes(l-1)}$ is diagonal and therefore only the diagonal unitaries from our set $U^{(mn)}$, i.e., $U^{(m0)}$, appear in its expansion, Eq. (3.47). Moreover, the coefficients q_{m0} in the expansion are non-vanishing only for odd m. It follows that

$$U_\varphi = \cos\varphi\, I^{\otimes l} + i \sin\varphi \sum_{\text{odd } m} q_{m0} U^{(m0)}, \tag{3.66}$$

and the probability of a successful implementation of this unitary transformation is $p = 2/(2^l + 2)$.

Example 2.

For some sets of operators it is possible to do even better than we were able to do in the previous example. Consider the one-parameter set of unitary operators given by

$$U_\vartheta = \cos\vartheta\, I + i \sin\vartheta\, U^{(0,N/2)}, \tag{3.67}$$

where N is assumed to be even. That this operator is unitary follows from the fact that $U^{(0,N/2)}$ is self-adjoint. A program vector that would implement this operator is

$$|\Phi\rangle_{23} = \cos\vartheta\,|\Xi_{00}\rangle_{23} + i\sin\vartheta\,|\Xi_{0,N/2}\rangle_{23}, \tag{3.68}$$

and at the output of the program register we make a projective measurement corresponding to the vector

$$|M\rangle_{23} = \frac{1}{\sqrt{2}}(|\Xi_{00}\rangle_{23} + |\Xi_{0,N/2}\rangle_{23}. \tag{3.69}$$

The probability for successfully achieving the desired result, i.e., the vector $U_\vartheta|\Psi\rangle_1$ in the data register, is $1/2$ irrespective of the value N, i.e., the number of qubits.

Comments

Above we have presented a programmable quantum processor that exactly implements a set of operators that form a basis for the space of operators on qudits. This processor has a particularly simple representation in terms of elementary quantum gates. It is, however, by no means unique. It is possible, in principle, to build a processor that exactly implements any set of unitary operators that form a basis for the set of operators on qudits of dimension N, and uses any orthonormal set of N^2 vectors as programs. Explicitly, if the set of operators is $\{V_n|n=1,\ldots N^2\}$ and the program vectors are $\{|y_n\rangle|n=1,\ldots N^2\}$, the processor transformation is given by

$$P_{dp} = \sum_{n=1}^{N^2} V_n^{(d)} \otimes |y_n\rangle_p\,{}_p\langle y_n|, \tag{3.70}$$

where the superscript (d) on the operator V_n indicates that it acts on the data register.

As an example, consider a data register consisting of l qubits. We could use the processor discussed in Sec. 3 to perform operations on states in this register, but we can also do something else; we can use l single-qubit processors, one for each qubit of the data register. Specifically, our unitary basis for the set operations on the data register would be

$$U_{JK} = U_{j_1k_1,\ldots,j_lk_l} = \bigotimes_{m=1}^{l} S_{j_mk_m} \tag{3.71}$$

where $J=(j_1,\ldots,j_l)$ and $K=(k_1\ldots,k_l)$ are sequences of zeros and ones, and the operators $S_{j_mk_m}$ are the defined immediately after Eq. (3.10). The

program register would consist of l pairs of qubits, $2l$ qubits in all, with each pair controlling the operation on one of the qubits in the data register. Each of the operators in our basis can be implemented perfectly by a program consisting of the tensor product state, $\prod_{m=0}^{l} |\Xi_{j_m k_m}^{(m)}\rangle$, where $|\Xi_{j_m k_m}^{(m)}\rangle$ is a two-qubit state that implements the operation $S_{j_m k_m}$ on the mth qubit of the data register.

We are then faced with the problem of which processor to use. This very much depends on the set of operations we want to apply to the data. How to choose the processor so that a given set of operations can be implemented with the greatest probability, for a fixed size of the program register is an open problem. A second issue is simplicity. One would like the processor itself and the program states it uses to be as simple as possible. The simplicity of the processor is related to the number of quantum gates it takes to construct it. We would maintain that the processors we have presented here are simple, though whether there are simpler ones we do not know. Judging the simplicity of the program states is somewhat more difficult, but they should be related in a relatively straightforward way to the operation that they encode. In many cases these states will have been produced by a previous part of a quantum algorithm, and complicated program states will mean more complexity for the algorithm that produces them. The program states proposed by Vidal, Masanes and Cirac (VMC) and the ones proposed by us in Sec. 2 are, in our opinion, simple.

In the following section we will analyze how one can improve a the probability of successfully carrying out a set of operations by increasing the dimensionality of the space of program vectors. VMC showed how to do this in a particular case, but more general constructions would be desirable.[11] Doing so we introduce a method of designing programs for a quantum computer.

4. IMPROVING THE PERFORMANCE OF PROBABILISTIC PROGRAMMABLE PROCESSORS

In a probabilistic processor, one measures the output program state. If the proper result is obtained, the desired operation has been performed on the data state, and if not, then the output of the data register is discarded. In this kind of a scenario, one wants the probability of successfully performing the operation to be as close to one as possible. In fact, what one would like, is, given a set of operations that one wishes to perform, a procedure for systematically increasing the probability of successfully performing these operations.

As we have already discussed, in the case of one-parameter unitary groups acting qubits this was done Preskill[10] and VMC.[11] considered the one-parameter group of operations given by $U(\theta) = \exp(i\theta\sigma_z)$, for $0 \leq \theta < 2\pi$, and discussed two equivalent methods of making the probability of performing $U(\theta)$ arbitrarily close to one. A circuit consisting of a single C-NOT gate, with the control qubit as the data and the target qubit as the program, can successfully perform $U(\theta)$ with a probability of 1/2. If the procedure fails, however, the data qubit, which was initially in the state $|\psi\rangle$, is left in the state $U(-\theta)|\psi\rangle$. What we can now do, is to send this qubit back into the same circuit, but with the program state that encodes the operation $U(2\theta)$. This also has a probability of 1/2 of succeeding, and increases the total success probability for the two-step procedure to 3/4. Note that our program state has increased to two qubits, one for the first step and one for the second. We can continue in this way simultaneously increasing the success probability and the size of the program state. It is also possible to design more complicated circuits that perform the entire procedure at once, i.e., they have a one-qubit data state, an N-qubit program state, and a success probability of $1 - (1/2)^N$.[11]

In this section we would like to extend these ideas in a number of different directions. First, we shall show that it is possible to boost the probability of sets of nonunitary operations. It will then be shown how to increase the success probability of operations on qudits. Finally, more complicated groups of operations will be considered.

4.1. Improving Operations on Qubits

We shall begin by describing the methods developed in Refs. 10 and 11 in terms of the formalism presented in Ref. 12. There, the input data state is in the Hilbert space $\mathcal{H}_d$, the program state in the space $\mathcal{H}_p$, and G is the unitary operator, acting on the space $\mathcal{H}_d \otimes \mathcal{H}_p$, that describes the action of the circuit. This operator can be expressed as

$$G = \sum_{j,k=0}^{N} A_{jk} \otimes |j\rangle_p \, {}_p\langle k|, \tag{4.1}$$

where N is the dimension of $\mathcal{H}_p$, A_{jk} is an operator on $\mathcal{H}_d$, and $\{|j\rangle | j = 1, \ldots N\}$ is an orthonormal basis for the program space. The operators A_{jk} satisfy [12]

$$\sum_{j=1}^{N} A_{jk_1}^{\dagger} A_{jk_2} = \sum_{j=1}^{N} A_{k_1 j} A_{k_2 j}^{\dagger} = I_d \delta_{k_1 k_2}, \tag{4.2}$$

where I_d is the identity operator on $\mathcal{H}_d$. If the circuit acts on the input state $|\psi\rangle_d \otimes |\Xi\rangle_p$, we find that

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = \sum_{j=1}^{N} A_j(\Xi)|\psi\rangle_d \otimes |j\rangle_p, \tag{4.3}$$

where

$$A_j(\Xi) = \sum_{k=1}^{N} {}_p\langle k|\Xi\rangle_p A_{jk}. \tag{4.4}$$

Let us begin by using this formalism, let us look at a C-NOT gate and the simplest of the circuits discussed in Ref. 11. Both the data and program space are two-dimensional, and the data space is the control qubit and the program space is the target qubit. Expressing the operator for the C-NOT gate in the form given in Eq. (4.1), and choosing the basis $\{|0\rangle, |1\rangle\}$ for the program space, we find that

$$\begin{aligned} A_{00} &= |0\rangle\langle 0|; \quad A_{01} = |1\rangle\langle 1|; \\ A_{10} &= |1\rangle\langle 1|; \quad A_{11} = |0\rangle\langle 0|. \end{aligned} \tag{4.5}$$

We want to use this circuit to perform the operation $U(\theta)$ and this can be done with the program state

$$|\Xi(\theta)\rangle = \frac{1}{\sqrt{2}}(e^{i\theta}|0\rangle + e^{-i\theta}|1\rangle). \tag{4.6}$$

This gives us the output state

$$G(|\psi\rangle_d \otimes |\Xi(\theta)\rangle_p) = \sum_{j=0}^{1} A_j(\theta)|\psi\rangle_d \otimes |j\rangle_p \tag{4.7}$$

where the program operators are

$$\begin{aligned} A_0(\theta) &= \frac{e^{i\theta}}{\sqrt{2}}|0\rangle\langle 0| + \frac{e^{-i\theta}}{\sqrt{2}}|1\rangle\langle 1| = \frac{1}{\sqrt{2}}U(\theta) \\ A_1(\theta) &= \frac{e^{i\theta}}{\sqrt{2}}|1\rangle\langle 1| + \frac{e^{-i\theta}}{\sqrt{2}}|0\rangle\langle 0| = \frac{1}{\sqrt{2}}U(-\theta). \end{aligned} \tag{4.8}$$

Therefore, if we measure the output of the program register in the computational basis and obtain $|0\rangle$, then $U(\theta)$ has been carried out on the data state. This occurs with a probability of 1/2.

If we obtain $|1\rangle$ instead of $|0\rangle$ when we measure the program register output, then the operation $U(-\theta)$ has been performed on the data state. We can try to correct this by sending the state $U(-\theta)|\psi\rangle_d$ back into the same circuit, but with the program state $|\Xi(2\theta)\rangle_p$. If we measure the program output and obtain $|0\rangle$, then the output of the data register is

$$U(2\theta)U(-\theta)|\psi\rangle_d = U(\theta)|\psi\rangle_d, \tag{4.9}$$

and this happens with a probability of 1/2. This will correct the previous error.

A circuit that does this all at once can be constructed from three qubits and two quantum gates.[11] Qubit 1 is the data qubit, and qubits 2 and 3 are the program qubits. The first gate is a C-NOT gate with qubit 1 as the control and qubit 2 as the target. The second gate is a Toffoli gate with qubits 1 and 2 as controls and qubit 3 as the target. A Toffoli gate does nothing to the control bits, and does nothing to the target bit unless both control bits are 1, in which case it flips the target bit. If we denote the orthonormal program space basis by

$$\begin{aligned} |0\rangle_p &= |0\rangle_2|0\rangle_3; \quad & |2\rangle_p &= |1\rangle_2|0\rangle_3; \\ |1\rangle_p &= |0\rangle_2|1\rangle_3; \quad & |3\rangle_p &= |1\rangle_2|1\rangle_3, \end{aligned} \tag{4.10}$$

then this circuit can be described by the operators

$$\begin{array}{llll} A_{00} = |0\rangle\langle 0|; & A_{01} = 0; & A_{02} = |1\rangle\langle 1|; & A_{03} = 0; \\ A_{10} = 0; & A_{11} = |0\rangle\langle 0|; & A_{12} = 0; & A_{13} = |1\rangle\langle 1|; \\ A_{20} = 0; & A_{21} = |1\rangle\langle 1|; & A_{22} = |0\rangle\langle 0|; & A_{23} = 0; \\ A_{30} = |1\rangle\langle 1|; & A_{31} = 0; & A_{32} = 0; & A_{33} = |0\rangle\langle 0|. \end{array} \tag{4.11}$$

The program state is now

$$|\Xi(\theta)\rangle = \frac{1}{2}\sum_{j=0}^{3} e^{i(3-2j)\theta}|j\rangle_p. \tag{4.12}$$

At the output of the processor the program register is measured in the computational basis, and only if both qubits are found to be in the state $|1\rangle$ does the procedure fail. The overall probability of succeeding is 3/4.

Now let us go back to the C-NOT gate with a single qubit program and consider a more general program state

$$|\Xi\rangle = c_0|0\rangle + c_1|1\rangle, \tag{4.13}$$

the operators $A_0(\Xi)$ and $A_1(\Xi)$ are

$$\begin{aligned} A_0(\Xi) &= c_0|0\rangle\langle 0| + c_1|1\rangle\langle 1|\,; \\ A_1(\Xi) &= c_1|0\rangle\langle 0| + c_0|1\rangle\langle 1|. \end{aligned} \tag{4.14}$$

These operators are not unitary, but they do have the property that $A_0(\Xi)A_1(\Xi) = A_1(\Xi)A_0(\Xi) = c_0c_1 I$. The output state of this circuit is given by Eq. (4.7), so that it can be used to realize, probabilistically, either of the nonunitary operators, $A_0(\Xi)$ or $A_1(\Xi)$. It also suggests that we should be able to apply something like the Preskill–Vidal–Masanes–Cirac scheme. In particular, suppose we are trying to perform the operation

$$B(z) = |0\rangle\langle 0| + z|1\rangle\langle 1|. \tag{4.15}$$

If $c_1 = zc_0$, then $A_0(\Xi)$ is proportional to $B(z)$. We send the data state into the processor and then measure the program state in the $\{|0\rangle, |1\rangle\}$ basis. If we get 0 we have succeeded, but if we get 1 we have instead applied $A_1(\Xi)$ to the state. If we fail, however, we can try again. We now take the output from our first attempt, which is $A_1(\Xi)|\psi\rangle_d$, and send it into the processor again, but this time with the program state

$$|\Xi'\rangle = \left(\frac{1}{1+|z|^4}\right)^{1/2} (|0\rangle + z^2|1\rangle)\,. \tag{4.16}$$

We again measure the program state, and if we find 0, the output of the data register is the desired state, $A_0(\Xi)|\psi\rangle_d$. If we failed, that is we found 1, we can try yet again, but we need to modify the program state every time we repeat the process.

Rather than performing this procedure sequentially, i.e., sending in the input state, seeing if we succeed, and if not trying the procedure again with a modified program state, we can again do everything at once by *enlarging* the size of the program space. We shall use a slightly different processor than the one used by Vidal and Cirac. It has the same four-dimensional program space, but the operators A_{jk} are now given by

$$\begin{array}{llll} A_{00} = |0\rangle\langle 0|; & A_{01} = |1\rangle\langle 1|; & A_{02} = 0; & A_{03} = 0; \\ A_{10} = 0; & A_{11} = |0\rangle\langle 0|; & A_{12} = |1\rangle\langle 1|; & A_{13} = 0; \\ A_{20} = 0; & A_{21} = 0; & A_{22} = |0\rangle\langle 0|; & A_{23} = |1\rangle\langle 1|; \\ A_{30} = |1\rangle\langle 1|; & A_{31} = 0; & A_{32} = 0; & A_{33} = |0\rangle\langle 0|. \end{array} \tag{4.17}$$

The program state is now

$$|\Xi\rangle_p = \sum_{k=0}^{3} c_k |k\rangle_p\,, \tag{4.18}$$

where $c_{k+1} = z c_k$ for $k = 0, 1, 2$, and normalization then requires that

$$|c_0|^2 = \frac{1 - |z|^2}{1 - |z|^8}. \tag{4.19}$$

The operation of the processor is given by

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = \sum_{j=0}^{3} A_j(\Xi)|\psi\rangle_d \otimes |j\rangle_p, \tag{4.20}$$

where

$$A_j(\Xi) = \sum_{k=0}^{3} c_k A_{jk}, \tag{4.21}$$

and the operators A_{jk} are given in Eq. (4.17). This processor will perform the operation $B(z)$ with a reasonably high probability. In order to see this, we first note that $A_j(\Xi) = z^j A_0(\Xi)$ for $j = 0, 1, 2,$. This implies that

$$\begin{aligned} G(|\psi\rangle_d \otimes |\Xi\rangle_p) = {} & A_0(\Xi)|\psi\rangle_d \otimes \left(\sum_{j=0}^{2} z^j |j\rangle_p \right) \\ & + A_3(\Xi)|\psi\rangle_d \otimes |3\rangle_p, \end{aligned} \tag{4.22}$$

and $A_0(\Xi) = c_0 B(z)$. At the output of the processor we measure the program state in the $\{|j\rangle | j = 0, \ldots 3\}$ basis, and if we get $0, 1$ or 2, we have carried out the desired operation. If $|\psi\rangle_d = \alpha|0\rangle + \beta|1\rangle$, then the probability of success depends on the input state and is given by

$$P_{\text{suc}} = \left(\frac{1 - |z|^6}{1 - |z|^8} \right) (|\alpha|^2 + |z|^2 |\beta|^2). \tag{4.23}$$

If we average this probability over all input states we find that

$$\overline{P}_{\text{suc}} = \frac{1}{2} \left(\frac{1 - |z|^6}{1 - |z|^8} \right) (1 + |z|^2). \tag{4.24}$$

As an example, we can consider the case $|z|^2 = 1/2$, which gives us $\overline{P}_{\text{suc}} = 0.7$.

This can easily be generalized to an N-dimensional program. The operators A_{jk} are now given by

$$A_{jk} = \delta_{j,k}|0\rangle\langle 0| + \delta_{j+1,k}|1\rangle\langle 1|, \tag{4.25}$$

where the addition in the second Kronecker delta is done modulo N. These operators satisfy Eq. (4.2), so that they define a unitary operator. The program state is now

$$|\Xi\rangle = c_0 \sum_{j=0}^{N-1} z^j |j\rangle_p, \tag{4.26}$$

where

$$|c_0|^2 = \frac{1-|z|^2}{1-|z|^{2N}}. \tag{4.27}$$

This yields the following output state

$$\begin{aligned} G(|\psi\rangle_d \otimes |\Xi\rangle_p) = c_0 B(z)|\psi\rangle_d \otimes \sum_{j=0}^{N-2} z^j |j\rangle_p \\ + A_{N-1}(\Xi)|\psi_d\rangle \otimes |N-1\rangle_p, \end{aligned} \tag{4.28}$$

where

$$A_{N-1}(\Xi) = c_0(z^{N-1}|0\rangle\langle 0| + |1\rangle\langle 1|). \tag{4.29}$$

The probability of successfully performing $B(z)$ on $|\psi\rangle_d$ is given by

$$\begin{aligned} P_{\text{suc}} &= 1 - \|A_{N-1}(\Xi)\psi\|^2 \\ &= 1 - \frac{(1-|z|^2)(|\alpha|^2|z|^{2(N-1)} + |\beta|^2)}{|z|^{2N}-1}. \end{aligned} \tag{4.30}$$

When $|z| = 1$, this is equal to $1 - (1/N)$. An examination of P_{suc} shows that it is an increasing function of N. In the case that $|z| = 1$ it approaches 1 as $N \to \infty$. This is no longer true if $|z| \neq 1$; if $|z| < 1$, we find that the limit is

$$P_{\text{suc}} \to 1 - (1-|z|^2)|\beta|^2 = \|B(z)\psi\|^2, \tag{4.31}$$

and if $|z| > 1$, the limit is

$$P_{\text{suc}} \to 1 - \left(1 - \frac{1}{|z|^2}\right)|\alpha|^2 = \frac{1}{|z|^2}\|B(z)\psi\|^2. \tag{4.32}$$

Therefore, only in the case that we are implementing a unitary operation can this sequence of processors achieve a success probability arbitrarily close to 1.

4.2. Improving Performance of Qudit Processors

We now want to see how these arguments can be generalized to higher dimensional systems, and, for the sake of simplicity, let us start by examining qutrits. The data space is now three-dimensional, and let us take for the operators A_{jk}

$$\begin{aligned} A_{00} &= |0\rangle\langle 0|; \quad A_{01} = |1\rangle\langle 1|; \quad A_{02} = |2\rangle\langle 2|; \\ A_{10} &= |2\rangle\langle 2|; \quad A_{11} = |0\rangle\langle 0|; \quad A_{12} = |1\rangle\langle 1|; \\ A_{20} &= |1\rangle\langle 1|; \quad A_{21} = |2\rangle\langle 2|; \quad A_{22} = |0\rangle\langle 0|. \end{aligned} \tag{4.33}$$

The general program state is

$$|\Xi\rangle = c_0|0\rangle + c_1|1\rangle + c_2|2\rangle, \tag{4.34}$$

which gives the program operators

$$\begin{aligned} A_0(\Xi) &= c_0|0\rangle\langle 0| + c_1|1\rangle\langle 1| + c_2|2\rangle\langle 2|; \\ A_1(\Xi) &= c_0|2\rangle\langle 2| + c_1|0\rangle\langle 0| + c_2|1\rangle\langle 1|; \\ A_2(\Xi) &= c_0|1\rangle\langle 1| + c_1|2\rangle\langle 2| + c_2|0\rangle\langle 0|. \end{aligned} \tag{4.35}$$

The output state is

$$|\Psi\text{out}\rangle = \sum_{j=0}^{2} A_j|\psi\rangle_d \otimes |j\rangle_p, \tag{4.36}$$

so that if we measure in the program space and get j, the output state of the data register is $A_j(\Xi)|\psi\rangle_d$.

Suppose we are trying to apply the operator $A_0(\Xi)$ to the input data state. The probability of succeeding is $\langle\psi|A_0^\dagger(\Xi)A_0(\Xi)|\psi\rangle$. If we fail, however, we can try again, and this will increase the total probability of success. To see how this works, let us consider an example. Suppose that we measured the program register and got 1 instead of 0. That means we now have the state $A_1(\Xi)|\psi\rangle_d$. We take this state and put it through the processor again, but with a modified program state

$$|\Xi'\rangle = c_0'|0\rangle + c_1'|1\rangle + c_2'|2\rangle. \tag{4.37}$$

Suppose we now measure the output in the program space and get 0. If $A_0(\Xi')A_1(\Xi) \propto A_0(\Xi)$, then we have succeeded on our second try. Noting that

$$A_0(\Xi') = c_0'|0\rangle\langle 0| + c_1'|1\rangle\langle 1| + c_2'|2\rangle\langle 2|, \tag{4.38}$$

we see that this condition is satisfied if

$$c_0' = \frac{\alpha c_0}{c_1}; \quad c_1' = \frac{\alpha c_1}{c_2}; \quad c_2' = \frac{\alpha c_2}{c_0}. \tag{4.39}$$

The constant α is chosen so that $|\Xi'\rangle$ is normalized.

What we can conclude from this is that we can, by trial and correction, boost the probabilities of implementing operators that are *diagonal* in the basis $\{|0\rangle, |1\rangle, |2\rangle\}$. In the case that the operator we are trying to implement is unitary, i.e., $|c_j| = 1/\sqrt{3}$, then our probability of success at each trial is 1/3, so that our probability of success after N trials is $1-(2/3)^N$. This probability goes to 1 as N goes to infinity. These conclusions generalize in a straightforward way to qudits.

We now want to explore increasing the probability of successfully performing an operation on qudits by increasing the size of the program space. The data space is now of dimension D, and the orthonormal basis spanning it is $\{|0\rangle_d, \ldots |D-1\rangle_d\}$. We shall consider a particular kind of operation, one that changes the amplitude of one of the basis states, and leaves the rest alone (up to overall normalization). Suppose the state whose amplitude we want to change is $|0\rangle_d$. The operator we want to implement is

$$B_0(z) = z|0\rangle_p\,{}_p\langle 0| + X, \tag{4.40}$$

where

$$X = \sum_{k=1}^{D-1} |k\rangle_p\,{}_p\langle k|. \tag{4.41}$$

For our processor, we shall choose the operators A_{jk}, where j and k run from 0 to $D-1$ to be

$$A_{jk} = \delta_{jk} X + \delta_{k,j+1} |0\rangle_p\,{}_p\langle 0|, \tag{4.42}$$

were all additions are modulo D. The program state

$$|\Xi\rangle_p = c_0 \sum_{k=0}^{N-1} z^k |k\rangle_p, \tag{4.43}$$

where $|c_0|^2$ is given by Eq. (4.27), gives us, for $0 \le j \le N-2$

$$A_j(\Xi) = c_0 z^j B_0(z). \tag{4.44}$$

The probability of successfully performing $B_0(z)$ on the data state $|\psi\rangle_d$, P_{suc}, is

$$P_{\text{suc}} = \frac{|z|^{2(N-1)} - 1}{|z|^{2N} - 1} \| B_0 \psi \|^2, \tag{4.45}$$

when $|z| \neq 1$, and it is $(N-1)/N$ when $|z| = 1$. In the limit that N goes to infinity, P_{suc} goes to one if $|z| = 1$. If $|z| > 1$ we have that

$$P_{\text{suc}} \to \frac{1}{|z|^2} \| B_0 \psi \|^2, \tag{4.46}$$

and if $|z| < 1$, then

$$P_{\text{suc}} \to \| B_0 \psi \|^2. \tag{4.47}$$

As before, we see that it is only in the case that the operation is unitary that the probability goes to one.

If we want to modify more than one basis vector amplitude, we can apply these processors successively, each designed to modify a single amplitude. In the case that all of the operations are unitary, this is a D-dimensional, programmable phase gate, whose probability of succeeding can be made arbitrarily close to one.

4.3. Realization of SU(2) Rotations

In the VMC model the angle of the U(1) rotation that is supposed to be performed on a qubit is encoded in a quantum state of the program. The rotation itself is then applied on the data qubit via the C-NOT gate that plays the role of a programmable processor. As we have discussed above the probability of success of the rotation can be enhanced, providing the data qubit is processed conditionally in loops. The dynamics of each "run" of the processor is conditioned by the result of the measurement performed on the program register.

In what follows we will show that an analogous strategy can be applied in the case of the SU(2) rotations of a qubit, when the parameters (angles) of the SU(2) rotations are encoded in the state of the program. In Sec. 3.2 we have shown an arbitrary single-qubit unitary transformation can be implemented with the probability $p = 1/4$ by using a quantum information distributor machine (QID) as the processor. As shown earlier the QID is a quantum processor with a single data qubit and two program qubits. The quantum information distribution is realized via a sequence of four C-NOT gates, such that firstly the data qubit controls

the NOT operation on the first and the second program qubits and then the first and the second program qubits act as the control with the data qubit as the target. At the end of this process a projective measurement on the two program qubits is performed. The measurement is performed in the basis: $\{|0\rangle|+\rangle; |0\rangle|-\rangle; |1\rangle|+\rangle; |1\rangle|-\rangle\}$ (where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$). The realization of the desired transformation is associated with the projection onto the vector $|0\rangle|+\rangle$. In what follows we will explicitly show how to correct the cases of wrong results, i.e., of projections onto one of the vectors $|0\rangle|-\rangle, |1\rangle|+\rangle, |1\rangle|-\rangle$.

Let us note that the action of the QID processor[12,24] can be expressed in the form

$$P = \sum_{j=0}^{3} \sigma_j \otimes |\Xi_j\rangle\langle\Xi_j| , \tag{4.48}$$

where σ_j are standard σ-matrices with $\sigma_0 = I$. The basis program vectors $|\Xi_j\rangle$ form the standard Bell basis, i.e.,

$$|\Xi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle); \qquad |\Xi_x\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle);$$
$$|\Xi_z\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle); \qquad |\Xi_y\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

The general program state $|\Xi(\vec{\mu})\rangle_p$ encoding the unitary transformation $U_{\vec{\mu}} = \exp(i\vec{\mu}.\vec{\sigma}) = \cos\mu I + i\sin\mu \frac{\vec{\mu}}{\mu}.\vec{\sigma}$ $(\mu = |\vec{\mu}|)$ is given by the expression

$$|\Xi(\vec{\mu})\rangle_p = \cos\mu|\Xi_0\rangle + i\frac{\sin\mu}{\mu}(\mu_x|\Xi_x\rangle + \mu_y|\Xi_y\rangle + \mu_z|\Xi_z\rangle). \tag{4.49}$$

Performing the previously mentioned measurement in the program basis $|0+\rangle, |0-\rangle, |1+\rangle, |1-\rangle$ we obtain the following unitary transformations

$$|0\rangle \otimes |+\rangle : |\psi\rangle_d \to U_{\vec{\mu}}|\psi\rangle_d ;$$
$$|0\rangle \otimes |-\rangle : |\psi\rangle_d \to \sigma_z U_{\vec{\mu}} \sigma_z |\psi\rangle_d ;$$
$$|1\rangle \otimes |+\rangle : |\psi\rangle_d \to \sigma_x U_{\vec{\mu}} \sigma_x |\psi\rangle_d ;$$
$$|1\rangle \otimes |-\rangle : |\psi\rangle_d \to \sigma_y U_{\vec{\mu}} \sigma_y |\psi\rangle_d ,$$

where

$$U_{\vec{\mu}} = \cos\mu I + \frac{i\sin\mu}{\mu}(\mu_x\sigma_x + \mu_y\sigma_y + \mu_z\sigma_z) . \tag{4.50}$$

To obtain this simple expression we have used the identity $\sigma_j\sigma_k\sigma_j = -\sigma_k$ if $k \neq j$. All observed outcomes occur with the same probability, $p = 1/4$. Using the above notation the action of the QID can be expressed in the form

$$|\psi\rangle_d \otimes |\Xi(\vec{\mu})\rangle_p \to \frac{1}{2}\left(\sum_{j=0}^{3} \sigma_j U_{\vec{\mu}} \sigma_j |\psi\rangle_d \otimes |\tilde{j}\rangle_p\right) \tag{4.51}$$

where vectors $\{|\tilde{j}\rangle_p\}$ form the basis of $\mathcal{H}_p$ associated with the realized measurement. The explicit form of the vectors is presented in following paragraph where we discuss a general solution of SU(N) rotations of qudits.

We see that each outcome of the measurement indicates a different unitary transformation that has been applied to the data. Once we have obtained a specific result we can use the same processor again to correct an incorrectly transformed data register and consequently improve the success probability. In particular, in the case of the result j, the new program register needs to encode the correcting transformation $U_j^{(1)} = U_{\vec{\mu}}\sigma_j U_{\vec{\mu}}^{\dagger}\sigma_j$. The probability of implementing the unitary transformation using one conditioned loop is given as $p(1) = \frac{1}{4} + 3\frac{1}{16} = \frac{7}{16}$. Using more and more conditioned loops the success probability is given by $p(n) = \sum_{j=1}^{n} \frac{1}{4^j} 3^{j-1} = 1/4 \sum_j (3/4)^j = \frac{1}{4}\frac{1-(3/4)^n}{1/4} = 1 - (3/4)^n$ converges to unity, i.e., $p(n) \to 1$ as the number of conditioned loops n goes to infinity. For instance, 30 conditioned loops result in the negligible probability of failure, $p_f \simeq 10^{-4}$.

4.4. SU (N) Rotations of Qudits

Now we will show that one can utilize the QID for a probabilistic implementation of SU(N) rotations of qudits. Following the action of the QID as the probabilistic processor (see Sec. 3.3) we perform a measurement on the program register in the basis

$$|\Phi_{rs}\rangle = \frac{1}{N}\sum_{m,n=0}^{N-1} \exp\left[2\pi i \frac{(mr - ns)}{N}\right] |\Xi_{mn}\rangle\,. \tag{4.52}$$

The orthogonality of this measurement basis directly follows from the orthogonality of the entangled basis $|\Xi_{mn}\rangle$. We should also note, that the vectors $|\Phi_{rs}\rangle$ itself can be rewritten in a factorized form, i.e.,

$$|\Phi_{rs}\rangle = |-r\rangle \otimes \frac{1}{\sqrt{N}}\sum_{n=0}^{N} \exp\left[2\pi i \frac{ns}{N}\right] |n - r\rangle\,, \tag{4.53}$$

which means that the measurement can be performed independently on two program qudits.

In order to clarify the role of the measurement we will rewrite the output state of the QID using the basis $|\Phi_{rs}\rangle$ for program qudits:

$$\begin{aligned} P_{123}|\Psi\rangle_1|\Xi_V\rangle_{23} &= \sum_{m,n=0}^{N-1} d_{m,n}U^{(m,n)}|\Psi\rangle_1|\Xi_{mn}\rangle_{23} \\ &= \sum_{m,n=0}^{N-1} d_{m,n}U^{(m,n)}|\Psi\rangle_1 \left[\frac{1}{N}\sum_{r,s=0}^{N-1}\exp\left[-2\pi i\frac{(mr-ns)}{N}\right]|\Phi_{rs}\rangle_{23}\right] \\ &= \frac{1}{N}\sum_{r,s=0}^{N-1}\sum_{m,n=0}^{N-1}\left\{\exp\left[-2\pi i\frac{(mr-ns)}{N}\right]d_{m,n}U^{(m,n)}\right\}|\Psi\rangle_1|\Phi_{rs}\rangle_{23}. \end{aligned} \tag{4.54}$$

Taking into account that

$$\left[U^{(p,q)}\right]^{\dagger}U^{(m,n)}U^{(p,q)} = \exp\left[2\pi i\frac{(mq-np)}{N}\right]U^{(m,n)} \tag{4.55}$$

and choosing $p=s$ and $q=r$ we find

$$\frac{1}{N}\mathrm{Tr}\left[\left(U^{(s,r)}\right)^{\dagger}\left(U^{(m,n)}\right)^{\dagger}U^{(s,r)}V\right] = \exp\left[-2\pi i\frac{(mr-ns)}{N}\right]d_{m,n}. \tag{4.56}$$

Finally, the output of the QID can be rewritten in the form

$$P_{123}|\Psi\rangle_1|\Xi_V\rangle_{23} = \frac{1}{N}\sum_{r,s=0}^{N-1}\left[U^{(s,r)}V\left(U^{(s,r)}\right)^{\dagger}\right]|\Psi\rangle_1|\Phi_{rs}\rangle_{23}, \tag{4.57}$$

from which it is clear that if the result of the measurement of the two program qudits is $|\Phi_{rs}\rangle_{23}$, then the system (data) is left in the state $\left[U^{(s,r)}V\left(U^{(s,r)}\right)^{\dagger}\right]|\Psi\rangle_1$. Obviously, if $s=r=0$, then the operator V is applied on the data qudit. The probability of this outcome is $1/N^2$. For all other results of the measurement the data qudit is left in the state given above. One can use these output states with a modified program state to improve the performance of the programmable processor. Specifically, we have to use the new program state $|\Xi_V^{(r,s)}\rangle$ that is chosen after taking into account the result of the previous measurement. This program state has first to "correct" the wrong realization of the operation V during the previous "run" of the processor and then apply (probabilistically), the original operation V. For this reason, the new program state has to perform the operation

$$V^{(r,s)} = V\left[U^{(s,r)} V \left(U^{(s,r)}\right)^{\dagger}\right]^{-1}. \tag{4.58}$$

This process of error correction (conditional loops) can be used K times and the technique of conditioned loops can be exploited in order to amplify the probability of success. Applying the processor K times the probability of a successful application of the desired SU(N) operation V reads $p(K) = 1 - (1 - 1/N^2)^K$.

Comments

Till now we have shown how to encode information about the quantum dynamics V to be performed on a quantum system (data register) in the state of another quantum system (program register). This information is stored in such a way that the program can be used to probabilistically perform the stored transformation on the data. Above we have analyzed systematically how to perform U(1) rotations of qubits and qudits and one-parameter families of nonunitary operations when the angle of rotation is encoded in states of quantum programs. In addition, we have shown how to increase the probability of success when the quantum processor is used in loops with updated program states. We have generalized the whole problem and we have shown that one can use a very simple quantum processor, the so called quantum information distributor, to perform arbitrary SU(2) rotations of qubits as well as SU(N) rotations of qudits using the probabilistic programmable processor with the quantum program register initially prepared in states that carry the information about the operation to be performed on the data. It is also possible to use enlarged programs to increase the probability of success without the use of loops. In this case the measurement performed on the program register has to be modified accordingly. We have shown that if the processor is used in loops with properly chosen program states one can improve the performance of the quantum programmable processor so that the probability of failure decreases exponentially with the number of program qudits that store the information about transformation on the data qudit.

In what follows we will analyze a simple model of a programmable processor with N identical copies of the program register. We will study how one can optimally use the information encoded in N copies of the program in order to optimally perform the desired operation on the data register. In order to make our investigation as transparent as possible we will study a simple $U(1)$ rotation of a single qubit around the axis z.

5. INCREASING THE PROBABILITY OF SUCCESS WITH MULTIPLE COPIES OF PROGRAM STATES

We have already shown that the probability of successfully carrying out the $U(1)$ operation on the data qubit can be increased through the enlargement of the program space. Specifically, in the VMC scheme, if the first operation failed, that is, we performed $U(-\theta)$ on the data state, we could attempt to correct this by performing the rotation $U(2\theta)$ on the wrongly transformed data state $U(-\theta)|\Psi\rangle_d$ and if that failed we could attempt to perform the transformation $U(4\theta)$ on the data state $U(-3\theta)|\Psi\rangle_d$, etc. The N-qubit program state $|\Xi_\theta^{(N)}\rangle_{\vec{p}}$ used for this iterative operation can be written as

$$\begin{aligned}|\Xi_\theta^{(N)}\rangle_{\vec{p}} &= |\Xi_{2^N\theta}\rangle_{p_1} \otimes |\Xi_{2^{N-1}\theta}\rangle_{p_2} \otimes \cdots \otimes |\Xi_\theta\rangle_{p_N} \\ &= \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} e^{-ij\theta} |j\rangle_{\vec{p}}, \end{aligned} \tag{5.1}$$

with $|j\rangle_{\vec{p}} = |j_N\rangle_{p_N} \otimes |j_{N-1}\rangle_{p_{N-1}} \cdots \otimes |j_1\rangle_{p_1}$, where j_l is the lth bit in the binary representation of j.

As discussed in Sec. 4 instead of using iteratively the C-NOT processor one can design a general quantum processor

$$G_{dp} = \sum_{j,k=1}^{2^N-1} A_{jk} \otimes |j\rangle_p \quad {}_p\langle k|, \tag{5.2}$$

where $\{|j\rangle_p| \;\; j = 0, \ldots, 2^N - 1\}$ is an orthonormal basis for the program space and the A_{jk} are operators acting on the data space such that:

$$\sum_{j=0}^{2^N-1} A_{xj}^\dagger A_{jy} = \sum_{j=0}^{2^N-1} A_{xj} A_{jy}^\dagger = I_d \delta_{xy}. \tag{5.3}$$

The result of the circuit on the combined data and program states input $|\Psi\rangle_d \otimes |\Xi\rangle_p \in \mathcal{H}_d \otimes \mathcal{H}_p$ can be expressed as:

$$G(|\Psi\rangle_d \otimes |\Xi\rangle_p) = \sum_{j=0}^{2^N-1} A_j(\Xi)|\Psi\rangle_d \otimes |j\rangle_p, \tag{5.4}$$

where the *program operators* $A_j(\Xi)$ are given by:

$$A_j(\Xi) = \sum_{k=0}^{2^N-1} {}_p\langle k|\Xi\rangle_p A_{jk}\,. \tag{5.5}$$

If the measurement of the program state returns $|n\rangle_p$, then Eq. (5.4) tells us that the operation $A_n(\Xi)$ has been carried out on the data state.

To perform the $U(1)$ operation with only one iteration of the processor in the HZB scheme (described in Section 4), we use the same program state as for the VMC scheme given by Eq. (5.1). The circuit (processor) is then determined by the operators

$$A_{jk} = \delta_{j,k}|0\rangle_{dd}\langle 0| + \delta_{j\oplus 1,k}|1\rangle_{dd}\langle 1| \tag{5.6}$$

with $\oplus$ indicating addition modulo 2^N. The program state is then measured and any result other than $|2^N - 1\rangle_p$ indicates success. The success probability for this circuit is the same as that for the VMC circuit and it reads:

$$p = 1 - \frac{1}{2^N}\,. \tag{5.7}$$

This is the highest possible success probability achievable from the starting state $|\Xi_\theta^{(N)}\rangle_p$ for a general probabilistic quantum processor.[11]

5.1. Using Multiple Copies of the Basic Program State

5.1.1. Iterative Process with Multiple Copies of the Program State $|\Xi_\theta\rangle$

Given that θ is not known, it is not clear how the program states for the improved schemes above might in general be produced deterministically given no prior knowledge of θ. General execution of $U(1)$ on a data qubit using a single program qubit and a C-NOT gate is known to be optimally achieved using the program state $|\Xi_\theta\rangle$ given by Eq. (1.6) (see Refs. 41 and 42), so assuming the availability of this state seems a reasonable minimal assumption. To increase the probability of success above 1/2 using just a C-NOT, we require more copies of this basic program state and, if the operation $U(-\theta)$ has been carried out, we can reprocess the data state with a new copy of $|\Xi_\theta\rangle$ and continue this process until the desired transformation has been executed or until the available program states are exhausted.[9] If N, the number of available copies of $|\Xi_\theta\rangle$, is an odd number (there is no benefit to using an even number of program states), the probability p of succeeding before running out of copies of $|\Xi_\theta\rangle$ is given by the expression

$$p = 1 - \frac{1}{2^N}\binom{N}{(N-1)/2}, \tag{5.8}$$

[9] This is analogous to the Markov process "Gambler's ruin," where the game is fair and the gambler has unlimited credit.

and, in the limit of large N:

$$p_{N\to\infty} = 1 - \sqrt{\frac{2}{\pi N}}. \tag{5.9}$$

5.1.2. Single-shot Process with Multiple Copies of the Program State $|\Xi_\theta\rangle$

The process can be carried out with one iteration of a larger gate array where we use an odd number of program qubits N so that our combined program and data state is:

$$|\psi\rangle_d \otimes |\Xi_\theta\rangle_p^{\otimes N} = \frac{|\psi\rangle_d}{\sqrt{2^N}} \otimes \sum_{j=0}^{2^N-1} e^{-i|j|\theta} |j\rangle_{\vec{p}}, \tag{5.10}$$

where $|j|$ is the Hamming weight of the binary representation of j and we use the same basis for the program space as previously. Putting $A_{kk} = |0\rangle_{dd}\langle 0|$ as before, we select the position of the terms $A_{jk} = |1\rangle_{dd}\langle 1|$ according to the Hamming weight of the j and k such that

$$|k| = |j| + 1 \tag{5.11}$$

to the largest extent possible so that Eq. (5.3) is obeyed and we can position the other terms arbitrarily so as to respect Eq. (5.3). Where we can give the A_{jk} values according to Eq. (5.11), measurement in the program basis will, up to global phase, ensure that the data qubit has been transformed by $U(\theta)$. The rows (values of j) where $A_{jk} = |1\rangle_{dd}\langle 1|$ are not positioned according to $|k| = |j| + 1$ indicate measurement outcomes where the desired transformation has not been carried out but instead a rotation through some negative multiple of θ has occurred. The number R of rows that cannot be created so that Eq. (5.11) is obeyed is given by:

$$R = \binom{N}{(N-1)/2}. \tag{5.12}$$

Each (incorrect) program operator corresponding to one of these rows has probability 2^{-N} so again the success probability is given by Eq. (5.8).[10]

[10] In this case, unlike the VMC and HZB schemes, the distribution of particular incorrect results can differ according to how the A_{jk} are selected, although the overall probability of success is unchanged.

5.1.3. Preprocessing

If we wish to use, from a starting state of multiple copies of $|\Xi_\theta\rangle$, the VMC or HZB schemes, we can process these copies to produce a state of the form given in Eq. (5.1) that can then be used as the program state for the VMC or HZB processors. The X-qubit program state $|\Xi_\theta^{(X)}\rangle_p$ can be probabilistically constructed from a minimum of $N = 2^X - 1$ copies of $|\Xi_\theta\rangle$, and so it is possible, by preprocessing these copies of $|\Xi_\theta\rangle$, to construct, with some probability, a state $|\Xi(\theta)_s\rangle_p$ where $s \leq X$. A preprocessing scheme that produces the same overall probability of success, in executing $U(\theta)$ on a data qubit, as the schemes presented above can be constructed by permuting the phases in $|\Xi_\theta\rangle^{\otimes 2^X - 1}$ and making a measurement in the computational basis, initially on $2^X - 1 - X = M$ of the qubits.

We give two specific examples, of preprocessing. Firstly, we will assume to have three identical program states $|\Xi_\theta\rangle^{\otimes 3}$. Then we will consider the case with seven identical program states, i.e., $|\Xi_\theta\rangle^{\otimes 7}$. Using 3 and 7 program state we can probabilistically prepare the program states $|\Xi_\theta^{(2)}\rangle_p$ and $|\Xi_\theta^{(3)}\rangle_p$, respectively. In the Appendix we will quote the result for general N.

Preprocessing with $|\Xi_\theta\rangle^{\otimes 3}$

We have that:

$$\begin{aligned} |\Xi_\theta\rangle^{\otimes 3} = \frac{1}{2\sqrt{2}}(&|000\rangle + e^{-i\theta}|001\rangle + e^{-i\theta}|010\rangle \\ &+ e^{-2i\theta}|011\rangle + e^{-i\theta}|100\rangle + e^{-2i\theta}|101\rangle \\ &+ e^{-2i\theta}|110\rangle + e^{-3i\theta}|111\rangle), \end{aligned} \tag{5.13}$$

in the computational basis. The states that can be constructed from this are $|\Xi_\theta^{(1)}\rangle$ and $|\Xi_\theta^{(2)}\rangle$ which are, up to global phase and in the computational basis:

$$|\Xi_\theta^{(1)}\rangle = \frac{1}{\sqrt{2}}(|0\rangle) + e^{-i\theta}|1\rangle) \tag{5.14}$$

and

$$\begin{aligned} |\Xi_\theta^{(2)}\rangle &= |\Xi_{2\theta}\rangle \otimes |\Xi_\theta\rangle \\ &= \frac{1}{2}(|00\rangle) + e^{-i\theta}|01\rangle + e^{-i2\theta}|10\rangle + e^{-3i\theta}|11\rangle). \end{aligned} \tag{5.15}$$

The state is permuted, which has the effect of reassigning the phases:

$$
\begin{aligned}
|\Xi_\theta\rangle^{\otimes 3} \mapsto & \frac{1}{2\sqrt{2}}(|000\rangle + e^{-i\theta}|001\rangle + e^{-2i\theta}|010\rangle \\
& + e^{-3i\theta}|011\rangle + e^{-i\theta}|100\rangle + e^{-2i\theta}|101\rangle \\
& + e^{-i\theta}|110\rangle + e^{-2i\theta}|111\rangle) \qquad (5.16) \\
= & \left(\frac{|0\rangle}{\sqrt{2}} \otimes |\Xi_\theta^{(2)}\rangle\right) \\
& + \left(\frac{e^{-i\theta}|1\rangle}{\sqrt{2}} \otimes \left(\frac{|0\rangle}{\sqrt{2}} \otimes |\Xi_\theta^{(1)}\rangle + \frac{|1\rangle}{\sqrt{2}} \otimes |\Xi_\theta^{(1)}\rangle\right)\right). \qquad (5.17)
\end{aligned}
$$

Equation (5.17) shows that a measurement on the first (leftmost in the right-hand-side of the previous equation) qubit would either give $|\Xi_\theta^{(2)}\rangle$ upon measurement outcome $|0\rangle$, or a state, on measurement outcome $|1\rangle$ which can be reduced to $|\Xi_\theta^{(1)}\rangle$, up to global phase, by measurement of the remaining leftmost qubit. Each of these final results occurs with probability 1/2 and so, using Eq. (5.7), we find that the overall probability of successfully executing the operation $U(\theta)$ following preprocessing of the state and then input of the outcome, as a program state, into a HZB or VMC process is 5/8, which is in fact the same as that for iterative or single-shot processing of the state $|\Xi_\theta\rangle^{\otimes 3}$ discussed above.

The preprocessing transformation (5.17) can be easily realized using a single `C-NOT` gate with the second qubit in Eq. (5.13) playing the role of a control with the first qubit acting as a target.

Preprocessing with $|\Xi_\theta\rangle^{\otimes 7}$

In considering the preprocessing of $|\Xi_\theta\rangle^{\otimes 7}$ we introduce a technique for permutation design that is helpful in describing the derivation of the general preprocessing procedure for $|\Xi_\theta\rangle^{\otimes N}$.

The starting point is the state:

$$
\begin{aligned}
|\Xi_\theta\rangle^{\otimes 7} &= \frac{1}{\sqrt{128}} \sum_{j=0}^{127} e^{-i|j|}|j\rangle \\
&= \frac{1}{\sqrt{128}} \sum_{p=0}^{15} |p\rangle \otimes \sum_{q=0}^{7} e^{-i(|q|+|p|)\theta}|q\rangle \qquad (5.18)
\end{aligned}
$$

and the procedure is to perform a permutation of the state so that measurement of the first four qubits in the computational basis will yield either $|\Xi_\theta^{(3)}\rangle$ or a state from which measurement of the one or two remaining leftmost qubits will yield $|\Xi_\theta^{(2)}\rangle$ or $|\Xi_\theta^{(1)}\rangle$, respectively, up to a global phase. The numbers of terms with each phase are given by

$-ik\theta$	0	$-i\theta$	$-2i\theta$	$-3i\theta$	$-4i\theta$	$-5i\theta$	$-6i\theta$	$-7i\theta$
m	1	7	21	35	35	21	7	1

and the aim is to allocate those phases to terms so that, upon measurement of the leftmost four qubits, the state is either projected into $|\Xi_\theta^{(3)}\rangle$ or else a state from which further measurement will project into $|\Xi_\theta^{(2)}\rangle$ or $|\Xi_\theta^{(1)}\rangle$ up to global phase. Noting that one set of the phases 0, $-i\theta$, $-2i\theta$, $-3i\theta$, $-4i\theta$, $-5i\theta$, $-6i\theta$, $-7i\theta$ are available, the permutation can be constructed so that the 4-qubit measurement outcome $|0\rangle$ in Eq. (5.18) is:

$$\begin{aligned}\frac{1}{4}|0\rangle \otimes \frac{1}{\sqrt{8}}\Big(&|0\rangle + e^{-i\theta}|1\rangle + e^{-2i\theta}|2\rangle + e^{-3i\theta}|3\rangle \\ &+ e^{-4i\theta}|4\rangle + e^{-5i\theta}|5\rangle + e^{-6i\theta}|6\rangle e^{-7i\theta}|7\rangle\Big) \\ &= \frac{1}{4}|0\rangle \otimes |\Xi_\theta^{(3)}\rangle. \end{aligned} \tag{5.19}$$

The following phases

$-ik\theta$	0	$-i\theta$	$-2i\theta$	$-3i\theta$	$-4i\theta$	$-5i\theta$	$-6i\theta$	$-7i\theta$
m	0	6	20	34	34	20	6	0

remain unassigned in the permutation. It can be seen that the terms associated with the 4-qubit measurement outcome $|1\rangle$ cannot constitute $|\Xi_\theta^{(3)}\rangle$, as the requisite phases have already been allocated to the terms associated with the measurement outcome $|0\rangle$. However, allocation of the phases $-i\theta$, $-2i\theta$, $-3i\theta$ and $-4i\theta$ and also $-3i\theta$, $-4i\theta$, $-5i\theta$ and $-6i\theta$ allows that the permutation can be designed such that the 4-qubit measurement outcome $|1\rangle$ is

$$\begin{aligned}\frac{1}{4}|1\rangle \otimes \frac{1}{\sqrt{8}}\Big(&e^{-i\theta}|0\rangle + e^{-2i\theta}|1\rangle + e^{-3i\theta}|2\rangle + e^{-4i\theta}|3\rangle \\ &+ e^{-3i\theta}|4\rangle + e^{-4i\theta}|5\rangle + e^{-5i\theta}|6\rangle + e^{-6i\theta}|7\rangle\Big) \\ &= \frac{1}{4}|1\rangle \otimes \frac{1}{\sqrt{2}}\left(|0\rangle \otimes e^{-i\theta}|\Xi_\theta^{(2)}\rangle + |1\rangle \otimes e^{-3i\theta}|\Xi_\theta^{(2)}\rangle\right). \end{aligned} \tag{5.20}$$

A further measurement of the leftmost remaining qubit will project the state of remaining qubits into $|\Xi_\theta^{(2)}\rangle$ up to a global phase of $e^{-i\theta}$ or $e^{-3i\theta}$. The remaining phases are

$-ik\theta$	0	$-i\theta$	$-2i\theta$	$-3i\theta$	$-4i\theta$	$-5i\theta$	$-6i\theta$	$-7i\theta$
m	0	5	19	32	32	19	5	0

.

The same allocation can be performed for the 4-qubit measurement outcomes $|2\rangle$ to $|6\rangle$. The remaining unallocated phases are

$-ik\theta$	0	$-i\theta$	$-2i\theta$	$-3i\theta$	$-4i\theta$	$-5i\theta$	$-6i\theta$	$-7i\theta$
m	0	0	14	22	22	14	0	0

and it is therefore possible to construct the permutation so that the measurement outcomes $|7\rangle$ to $|13\rangle$ are

$$\begin{aligned}\frac{1}{4}|j\rangle \otimes \frac{1}{\sqrt{8}} \Big(& e^{-2i\theta}|0\rangle + e^{-3i\theta}|1\rangle + e^{-4i\theta}|2\rangle + e^{-5i\theta}|3\rangle \\ & + e^{-2i\theta}|4\rangle + e^{-3i\theta}|5\rangle + e^{-4i\theta}|6\rangle + e^{-5i\theta}|7\rangle\Big) \\ = \frac{1}{4}|j\rangle \otimes & \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes e^{-2i\theta}|\Xi_\theta^{(2)}\rangle; \quad j = 7 \ldots 13. \end{aligned} \tag{5.21}$$

Any measurement on the leftmost remaining qubit projects into the state $e^{-2i\theta}|\Xi_\theta^{(2)}\rangle$. Finally, the remaining phases,

$-ik\theta$	0	$-i\theta$	$-2i\theta$	$-3i\theta$	$-4i\theta$	$-5i\theta$	$-6i\theta$	$-7i\theta$
m	0	0	0	8	8	0	0	0

are allocated to the 4-qubit measurement outcomes $|14\rangle$ and $|15\rangle$ like so

$$\begin{aligned}\frac{1}{4}|l\rangle \otimes \frac{1}{\sqrt{8}} \Big(& e^{-3i\theta}|0\rangle + e^{-4i\theta}|1\rangle + e^{-3i\theta}|2\rangle + e^{-4i\theta}|3\rangle \\ & + e^{-3i\theta}|4\rangle + e^{-4i\theta}|5\rangle + e^{-3i\theta}|6\rangle + e^{-4i\theta}|7\rangle\Big) \\ = \frac{1}{2}|14\rangle \otimes & \left(\frac{|0\rangle + |1\rangle + |2\rangle + |3\rangle}{2}\right) \otimes e^{-3i\theta}|\Xi_\theta^{(1)}\rangle. \end{aligned} \tag{5.22}$$

with $l = 14, 15$. A measurement of the two leftmost remaining qubits will project the remaining qubits into the state $e^{-3i\theta}|\Xi_\theta^{(1)}\rangle$. Thus, the permutation construction is complete and the overall, permuted state, $|\tilde{\Xi}_\theta\rangle_7$ is given by:

$$
\begin{aligned}
|\tilde{\Xi}_\theta\rangle_7 = & \frac{1}{4}|0\rangle \otimes |\Xi_\theta^{(3)}\rangle \\
& + \frac{1}{4}\sum_{k=1}^{7} |k\rangle \otimes \frac{1}{\sqrt{2}}\left(|0\rangle \otimes e^{-i\theta}|\Xi_\theta^{(2)}\rangle + |1\rangle \otimes e^{-3i\theta}|\Xi_\theta^{(2)}\rangle\right) \\
& + \frac{1}{4}\sum_{k=8}^{13} |k\rangle \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes e^{-2i\theta}|\Xi_\theta^{(2)}\rangle \\
& + \frac{1}{2}\sum_{k=14}^{15} |k\rangle \otimes \left(\frac{|0\rangle + |1\rangle + |2\rangle + |3\rangle}{2}\right) \otimes e^{-3i\theta}|\Xi_\theta^{(1)}\rangle. \qquad (5.23)
\end{aligned}
$$

The probability of the preprocessing procedure, following the 4-qubit measurement in the computational basis, producing the outcome $|\Xi_\theta^{(3)}\rangle$ is 1/16, that of producing outcome $|\Xi_\theta^{(2)}\rangle$ is 13/16 and that of producing outcome $|\Xi_\theta^{(1)}\rangle$ is 1/8. The overall probability, p, then, of achieving the rotation $U(\theta)$ from the starting state $|\Xi_\theta\rangle^{\otimes 7}$ by preprocessing and then input of the preprocessed state into the VMC or HZB processors, is

$$
p = \left(\frac{7}{8} \times \frac{1}{16}\right) + \left(\frac{3}{4} \times \frac{13}{16}\right) + \left(\frac{1}{2} \times \frac{1}{8}\right) = \frac{93}{128}, \qquad (5.24)
$$

which is the same as the iterative or single-shot procedures outlined above, as can be confirmed with use of Eq. (5.8). It should be noted that the permutation outlined above is not unique and that other permutations could be devised to achieve the same overall success probability.

Preprocessing with $|\Xi_\theta\rangle^{\otimes N}$

The equivalence of the iterative, single-shot and preprocessing schemes can be shown to be true in general for states of $N = 2^X - 1$, $X = 1, 2, \ldots$ copies of $|\Xi_\theta\rangle$, as described in Appendix, so that the overall success probability from a preprocessing of the state $|\Xi_\theta\rangle^{\otimes N}$ as described above, followed by input of the result of the preprocessing into a VMC or HZB processor, is the same as that in Eq. (5.8), i.e.,

$$
p = 1 - \frac{1}{2^N}\binom{N}{(N-1)/2}, \qquad (5.25)
$$

and thus we see that the use of the VMC or HZB schemes holds no advantage in terms of overall success probability when we are constrained to start with $|\Xi_\theta\rangle^{\otimes N}$.

Comments

If we have no reason to assume that previous operations have produced a program state $|\Xi_\theta^{(N)}\rangle_{\vec{p}}$, then it is reasonable to assume that we only have access to copies of the basic program state $|\Xi_\theta\rangle$; in this case there is no advantage, in terms of probability of success, in using the more sophisticated VMC and HZB schemes to execute the desired $U(1)$ operation because what we gain from those schemes we lose in producing the correct input program state. It appears that all strategies, in practice, give the same probability of success in executing the desired $U(1)$ rotation on a qubit. There may, however, be contextual advantages to the preprocessing scheme, for example, if the program state is to be teleported to a remote location before execution of the program; in this case, preprocessing means that the number of qubits to be transported is significantly lessened, which would be helpful if teleportation resources are scarce. On the other hand, if teleportation is unreliable but teleportation resources are not scarce, it might be better to teleport the copies of the basic program state as is, because the effect of losing a program qubit is not so great as in the case of sending the preprocessed states.

It is an open question as to whether a similar situation holds for the execution of the most general unitary operations on a qubit, the SU(2) operations.

6. QUANTUM SIMULATIONS AND PROCESSOR DESIGN

In the previous sections we have studied sets of superoperators that a given processor can perform. We would now like to turn the problem around and suppose that we have a given set of superoperators, and our aim is to construct a processor that will be able to execute them. We already know that it is impossible to find a processor that will perform all superoperators. In particular, if the set of superoperators we are trying to implement contains an uncountable set of unitary superoperators, then the set of superoperators cannot be performed by a single processor.

Here we will ask more modest question: Under what circumstances we are able to find a processor that will perform some one-parameter set of superoperators? In particular, suppose that we have the superoperators T_θ, where the parameter θ varies over some range, and that these operators have a Kraus representation $\{B_j(\theta)|j=1,\ldots,M\}$ such that

$$T_\theta[\rho] = \sum_{j=1}^{M} B_j(\theta)\rho B_j^\dagger(\theta). \tag{6.1}$$

Our aim is to find a unitary operator, G, and a set of program states $|\Xi(\theta)\rangle_p$ so that

$$T_\theta[\rho_d] = G(\rho_d \otimes |\Xi(\theta)\rangle_{p\,p}\langle\Xi(\theta)|)G^\dagger. \tag{6.2}$$

The operators $A_j(\Xi)$ that represent the action of the processor on the data states when the program state is $|\Xi\rangle$, are now functions of θ and we shall denote them as $A_j(\theta)$. Our processor then transforms the input data state ρ_d into the output state, $\rho_d^{(\mathrm{out})}$

$$\rho_d^{(\mathrm{out})} = \sum_{j=1}^{N} A_j(\theta)\rho_d A_j^\dagger(\theta). \tag{6.3}$$

We note that the operators $\{A_j(\theta)|j=1,\ldots,N\}$ also constitute a Kraus representation of the superoperator T_θ. The Kraus representation of a superoperator is not unique; any two different Kraus representations of the same superoperator, $\{B_j|j=1,\ldots,M\}$ and $\{C_j|j=1,\ldots,N\}$, where $N \geq M$, are related as follows,[10]

$$C_j = \sum_{k=1}^{N} U_{kj} B_k, \tag{6.4}$$

where U_{kj} is a unitary matrix. It is understood that if $N > M$, then zero operators are added to the set $\{B_j|j=1,\ldots,M\}$ so that the two sets of operators have the same cardinality.

In what follows we will study two single-qubit quantum channels, the phase-damping channel and the amplitude-damping channel. We will show that the former can be realized by a finite quantum processor, while the second cannot.

6.1. Examples

6.1.1. Phase-Damping Channel

The phase-damping channel is described by the map T_θ that is determined by the Kraus operators $B_1(\theta) = \sqrt{\theta}I$ and $B_2(\theta) = \sqrt{1-\theta}\sigma_z$, where both σ_z and I are unitary operators, and $0 \leq \theta \leq 1$. [2,13,14] Hence for the phase-damping map we find

$$T_\theta[\varrho_d] = \theta I\varrho_d I + (1-\theta)\sigma_z\varrho_d\sigma_z^\dagger, \tag{6.5}$$

where ϱ_d is the input qubit state. We can design the corresponding processor using Eq. (2.24), that is

$$G^{\text{phase}}|\phi\rangle_d \otimes |k\rangle_p = (U_k|\phi\rangle_d) \otimes |k\rangle_p \,, \tag{6.6}$$

where $k=1,2$ and $U_1=I, U_2=\sigma_z$. The program state in which the required transformation T_θ is encoded is given by $|\Xi(\theta)\rangle_p = \sqrt{\theta}|0\rangle_p + \sqrt{1-\theta}|1\rangle_p$. Note that in this case the program operators, $A_j(\theta)$, for $j=1,2$, are equal to the corresponding Kraus operators, i.e., $A_j(\theta) = B_j(\theta)$. Therefore, we can execute the entire one parameter set of superoperators T_θ merely by changing the program state we send into the processor, and the dimension of the program space is two.

6.1.2. Amplitude-Damping Channel

The amplitude-damping map S_θ is given by the Kraus operators $B_1(\theta) = |0\rangle\langle 0| + \sqrt{1-\theta}|1\rangle\langle 1|$ and $B_2(\theta) = \sqrt{\theta}|0\rangle\langle 1|$, where again, $0 \le \theta \le 1$. In designing a processor to realize this channel, we would again like to assume that the program operators are the same as the Kraus operators, $B_1(\theta)$ and $B_2(\theta)$. In this case, however, we have a problem. The program operators must satisfy Eq. (2.22), but

$$\begin{aligned}\sum_{j=1}^{2} B_j^\dagger(\theta_1) B_j(\theta_2) \\ = |0\rangle\langle 0| + (\sqrt{\theta_1\theta_2} + \sqrt{(1-\theta_1)(1-\theta_2)})|1\rangle\langle 1|,\end{aligned} \tag{6.7}$$

and the right-hand side of this equation is not, in general, proportional to the identity.

What we now must do is try to find a Kraus representation for this channel that does satisfy Eq. (2.22). In particular, we assume that

$$C_k(\theta) = \sum_{k=1}^{N} U_{kj}(\theta) B_j(\theta), \tag{6.8}$$

where $U(\theta)$ is an $N \times N$ unitary matrix, and $B_j(\theta) = 0$ for $j > 2$. In addition, we want

$$\sum_{j=1}^{N} C_j^\dagger(\theta_1) C_j(\theta_2) = f(\theta_1, \theta_2) I, \tag{6.9}$$

where $f(\theta_1, \theta_2)$ is a function whose magnitude is less that or equal to one. The operators $C_j(\theta)$ would then be candidates for the program operators,

$A_j(\theta)$. What we will show is that there is no Kraus representation with N finite that satisfies these conditions. Because the number of program operators is equal to the dimension of the program space, this will show that there is no finite quantum processor that can realize the family of superoperators that describes the amplitude-damping channel.

If Eq. (6.9) is to hold, then the coefficients of $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$ must be the same. Inserting the explicit expressions for $C_j(\theta)$ in terms of $B_1(\theta)$ and $B_2(\theta)$, this condition becomes

$$\left(1-\sqrt{(1-\theta_1)(1-\theta_2)}\right)\sum_{j=1}^{N} U_{1j}^*(\theta_1)U_{1j}(\theta_2) = \sqrt{\theta_1\theta_2}\sum_{j=1}^{N} U_{2j}^*(\theta_1)U_{2j}(\theta_2). \tag{6.10}$$

We can now make use of the fact that the rows of a unitary matrix constitute orthonormal vectors and the Schwarz inequality to show that the magnitude of the sum on the right-hand side of this equation is less than or equal to one. This give us that

$$\left|\sum_{j=1}^{N} U_{1j}^*(\theta_1)U_{1j}(\theta_2)\right| \leq \frac{\sqrt{\theta_1\theta_2}}{1-\sqrt{(1-\theta_1)(1-\theta_2)}}. \tag{6.11}$$

We now need the result that if $\{v_j | j=1,\ldots,N\}$ are vectors of length 1, and $|\langle v_j|v_k\rangle| < 1/(N-1)$, then $\{v_j | j=1,\ldots,N\}$ are linearly independent.[11] The proof is quite short, so we give it here. If the vectors are linearly dependent, then there are constants c_j, at least some of which are not zero, such that

$$\sum_{j=1}^{N} c_j|v_j\rangle = 0. \tag{6.12}$$

Taking the inner product of both sides with $|v_k\rangle$ we find that

$$|c_k| = \left|\sum_{j\neq k} c_j\langle v_k|v_j\rangle\right| < \frac{1}{N-1}\sum_{j\neq k}|c_j|. \tag{6.13}$$

Summing both sides of the above inequality over k gives us that

$$\sum_{k=1}^{N} |c_k| < \frac{1}{N-1} \sum_{k=1}^{N} \sum_{j \neq k} |c_j| = \sum_{k=1}^{N} |c_k|, \tag{6.14}$$

which is clearly impossible. Therefore, the vectors must be linearly independent.

This can now be applied to the first row of the unitary matrix $U(\theta)$, which we can think of as an N-component normalized vector, which we shall call $u_0(\theta)$. What we will show is that we can find arbitrarily many of these vectors whose inner products can be made arbitrarily small. The result in the previous paragraph then implies that these vectors are linearly independent, but this contradicts the fact that they lie in an N-dimensional space. Hence, there must be an infinite number of Kraus operators, and the program space must be infinite dimensional.

In order to study the inner products of the vectors $u_0(\theta)$ for different values of θ, we need to examine the function appearing on the right-hand side of Eq. (6.11)

$$g(\theta_1, \theta_2) = \frac{\sqrt{\theta_1 \theta_2}}{1 - \sqrt{(1-\theta_1)(1-\theta_2)}}. \tag{6.15}$$

Using the fact that if $0 \leq \theta \leq 1$, then $\sqrt{1-\theta} \leq 1 - (\theta/2)$, we have that for $0 \leq \theta_j \leq 1,\ j = 1, 2$

$$g(\theta_1, \theta_2) \leq \frac{2\sqrt{\theta_1 \theta_2}}{\theta_1 + \theta_2 - (\theta_1 \theta_2 / 2)}. \tag{6.16}$$

Finally, noting that for θ_1 and θ_2 between 0 and 1,

$$\frac{\theta_1 + \theta_2}{\theta_1 + \theta_2 - (\theta_1 \theta_2 / 2)} \leq \frac{4}{3}, \tag{6.17}$$

we see that

$$g(\theta_1, \theta_2) \leq \frac{8\sqrt{\theta_1 \theta_2}}{3(\theta_1 + \theta_2)}. \tag{6.18}$$

We can make use of this bound, if we choose, for any positive integer M, the sequence $\zeta_n = [1/(16M^2)]^n$, where $n = 1, \ldots$. If $\theta_1 = \zeta_n$ and $\theta_2 = \zeta_m$ where $m > n$, then

$$g(\theta_1, \theta_2) \leq \frac{8}{3} \frac{1}{(4M)^{m-n}}. \tag{6.19}$$

The vectors $\{u_0(\zeta_m)|m = 1, \ldots, M\}$ have pairwise inner products whose magnitudes are less than $1/M$, and, therefore, they are linearly independent. As these vectors have N components, if we choose $M > N$ we have a contradiction. This, as we stated before, implies that the number of Kraus operators is infinite, and that the amplitude-damping channel cannot be realized by a finite quantum processor.

Note: More on simulation of generators of Markovian processes with the help of programmable quantum processors can be found in a recent paper by Koniorczyk *et al.*[43]

7. QUANTUM MEASUREMENTS VIA PROGRAMMABLE MEASUREMENTS

In this section we will study how programmable processors can be utilized for implementation of generalized POVM measurements. We will study three particular problems. Firstly, we will study possible realizations of generalized quantum measurements on measurement-assisted programmable quantum processors. We focus our attention on the realization of von-Neumann measurements and informationally complete POVMs.

Secondly, we will show that it is possible to control the trade-off between information gain and disturbance in generalized measurements of qudits by utilizing the programmable quantum processor. We will show how one can perform a specific POVM that would allow to measure (reconstruct) a Husimi function of the input state of a qudit that is measured. The trade-off between the gain and the disturbance of the qudit is controlled by the initial state of ancillary system that acts as a program register for the quantum information distributor. We will show that trade-off fidelity does not depend on the initial state of the qudit.

Thirdly, we will describe a "programmable" quantum device that is able to perform a specific generalized measurement from a certain set of measurements depending on a quantum state of a "program register." The state of the program register sets the measurement device to perform a specific measurement. In particular, we study a situation when the programmable measurement device serves for the unambiguous discrimination between non-orthogonal states. The particular pair of states that can be unambiguously discriminated is specified by the state of a program qubit. The probability of successful discrimination is not optimal for all admissible pairs. However, for some subsets it can be very close to the optimal value.

7.1. Realization of POVMs Using Measurement-assisted Programmable Quantum Processors

General quantum measurements are formalized as POVM, i.e., sets of positive operators $\{F_k\}$ that fulfil the resolution to the identity, $\sum_k F_k = I$ (see, for instance, Refs. 2, 10, 15 and 16). From the general structure of quantum theory[15] it follows that each collection of such operators corresponds to a specific quantum measurement. However, the theory does not directly specify a particular physical realization of a given POVM. In what follows we will exploit the *measurement-assisted quantum processors* to perform POVMs.

The *Stinespring–Kraus theorem*[44] relates quantum operations (*linear completely positive trace-preserving maps*) with unitary transformations. In particular, any quantum operation $\mathcal{E}$ realized on the system A corresponds to a unitary transformation G performed on a larger system $A+B$, i.e.,

$$\mathcal{E}[\varrho] = \mathrm{Tr}_B[G\varrho \otimes \xi G^\dagger], \tag{7.1}$$

where ξ is a suitably chosen state of the ancillary system B and Tr_B denotes a *partial trace* over the ancilla B. The assignment $\mathcal{E} \mapsto (G, \xi)$ is one-to-many, because the dilation of the Hilbert space of a system A can be performed in many different ways. However, if we fix the transformation G, the states ξ of the ancillary system B control and determine quantum operations that are going to be performed on the system A. In this way one obtains a concept of a *programmable quantum processor*, i.e., a "piece of hardware" that take as an input a data register (system A) and a program register (system B). Here the state of the program register ξ encodes the operation $\varrho \to \varrho' = \mathcal{E}_\xi[\varrho]$ that is going to be performed on the data register.

In a similar way, any quantum generalized measurement (POVM), that is represented by a set of positive operators $\{F_j\}$, can be understood as a *von-Neumann measurement* performed on the larger system.[16] The von-Neumann measurements are those for which $F_j \equiv E_j$ are mutually orthogonal projectors, i.e., $E_j E_k = \delta_{jk} E_k$. The *Neumark theorem* (see, e.g., Ref. 14) states that for each POVM $\{F_j\}$ there exists a von-Neumann measurement $\{E_j\}$ on a larger Hilbert space $\mathcal{H}_{AB}$ and $\mathrm{Tr}\varrho F_j = \mathrm{Tr}[(\varrho \otimes \xi)E_j]$ for all ϱ, where ξ is some state of the system B. Moreover, it is always possible to choose a von-Neumann measurement such that $E_j = G^\dagger(I \otimes Q_j)G$ where G is a unitary transformation and Q_j are projectors defined on the system B. Using the cyclic property of a trace operation, i.e., $\mathrm{Tr}[(\varrho \otimes \xi)G^\dagger(I \otimes Q_j)G] = \mathrm{Tr}[G(\varrho \otimes \xi)G^\dagger(I \otimes Q_j)]$, we see that the von-Neumann measurement can be understood as a unitary transformation G

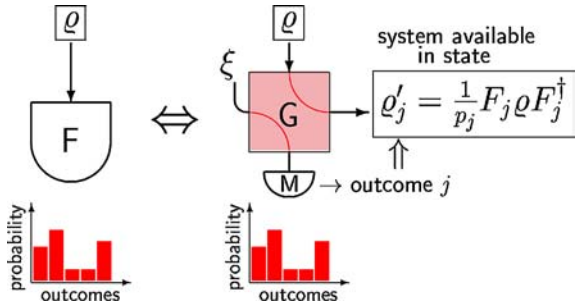

Fig. 8. With the help of a measurement-assisted quantum processor (the right part of the figure) one can realize an arbitrary POVM F (the left part of the figure) as a non-demolition measurement. After measuring the outcome j on the program register the system (i.e., the data register) is in the state ϱ_j. The correspondence between both schemes is given by the probability rule $p_j = \mathrm{Tr}\varrho F_j = \mathrm{Tr}[(I \otimes Q_j)G(\varrho \otimes \xi)G^\dagger]$.

followed by a von-Neumann measurement $M \leftrightarrow \{Q_j\}$ performed on the ancillary system only (see Fig. 8).

As a result we obtain the couple (G, M) that determines a programmable quantum processor assisted by a measurement of the program register, i.e., *measurement-assisted programmable quantum processor*. Such device can be used to perform both generalized measurements as well as quantum operations.

In this sub-section we will exploit measurement-assisted quantum processors to perform POVMs. In the next sub-section we will address problem of the implementation of a von-Neumann measurement by using programmable "quantum multimeters" for a discrimination of quantum state that has been first formulated in Ref. 45 and subsequently it has been studied in Refs. 46–48. An analogous setting of a unitary transformation followed by a measurement has been used in Ref. 49 to evaluate/measure the expectation value of any operator. The quantum network based on a controlled-SWAP gate can be used to estimate non-linear functionals of quantum states[50] without any recourse to quantum tomography. Recently D'Ariano and co-wrokers[51–53] have studied how programmable quantum measurements can be efficiently realized with finite-dimensional ancillary systems. In what follows we will study how von-Neumann measurements and informationally complete POVMs can be realized via programmable quantum measurement devices. In particular, we will show that this goal can be achieved using the quantum information distributor.[24,54]

7.1.1. General Consideration

Let us start our investigation with an assumption that the program register is always prepared in a pure state, i.e., $\xi = |\Xi\rangle\langle\Xi|$. In this case the action of the processor can be written as before, i.e.,

$$G|\psi\rangle \otimes |\Xi\rangle = \sum_k A_k(\Xi)|\psi\rangle \otimes |k\rangle\,, \tag{7.2}$$

where $|k\rangle$ is some basis in the Hilbert space of the program register and the operator $A_k(\Xi) = \langle k|G|\Xi\rangle$ act on the data register. In particular, we can use the basis in which the measurement M is performed, i.e., $Q_a = \sum_{k\in J_a} |k\rangle\langle k|$, where J_a is a subset of indices $\{k\}$. Note that $J_a \cap J_{a'} = \emptyset$, because $\sum_a Q_a = I$.

Measuring the outcome a the data evolve according to the following rule (*the projection postulate*)

$$\begin{aligned} \varrho \to \varrho'_a &= \frac{1}{p_a}\mathrm{Tr}_p[(I \otimes Q_a)G(\varrho \otimes |\Xi\rangle\langle\Xi|)G^\dagger] \\ &= \frac{1}{p_a}\sum_{k\in J_a} A_k(\Xi)\varrho A_k^\dagger(\Xi)\,, \end{aligned} \tag{7.3}$$

with the probability $p_a = \mathrm{Tr}[(I \otimes Q_a)G(\varrho \otimes |\Xi\rangle\langle\Xi|)G^\dagger] = \mathrm{Tr}[\varrho \sum_{k\in J_a} A_k^\dagger(\Xi)A_k(\Xi)] = \mathrm{Tr}[\varrho F_a]$. Consequently for the elements of the POVM we obtain

$$F_a = \sum_{k\in J_a} A_k^\dagger(\Xi)A_k(\Xi)\,. \tag{7.4}$$

If we consider a general program state with its spectral decomposition in the form $\xi = \sum_n \pi_n |\Xi_n\rangle\langle\Xi_n|$, then the transformation reads

$$\varrho \to \varrho'_a = \frac{1}{p_a}\sum_{n,k\in J_a} \pi_n A_{kn}\varrho A_{kn}^\dagger\,, \tag{7.5}$$

with $A_{kn} = \langle k|G|\Xi_n\rangle$ and $p_a = \sum_{n,k\in J_a} \pi_n \mathrm{Tr}[\varrho A_{kn}^\dagger A_{kn}]$. Therefore the operators

$$F_a = \sum_{n,k\in J_a} \pi_n A_{kn}^\dagger A_{kn} \tag{7.6}$$

constitute the realized POVM.

Given a processor G and some measurement M one can easily determine which POVM can be performed. Note that the same POVM can be realized in many physically different ways. Two generalized measurements

M_1, M_2 are equivalent, if the resulting functionals $f_k^{(x)}(\varrho) = \mathrm{Tr}\varrho F_k^{(x)}$ $(x = 1, 2)$ coincide for all k, i.e., they result in the same probability distributions. For the purpose of the realization of POVMs, the state transformation during the process is irrelevant. However, two equivalent realizations of POVM can be distinguished by the induced state transformations (for more on quantum measurement see Ref. 16).

Let us consider, for instance, the trivial POVM, which consists of operators $F_k = c_k I$ $(c_k \geq 0, \sum_k c_k = 1)$. In this case the observed probability distribution is data-independent and some quantum operation is realized. In all other cases, the state transformation depends on the initial state of the data register, and is not linear.[41,42]. In these cases the resulting distribution is nontrivial and contains some information about the state ϱ. In the specific case when the state ϱ can be determined (reconstructed) perfectly, the measurement is *informationally complete*. In this case we can perform the *complete state reconstruction* (see Fig. 9). Any collection of d^2 linearly independent positive operators F_k determine such informationally complete POVM. In particular, they form an operator basis, i.e., any state ϱ can be written as a linear combination $\varrho = \sum_j \varrho_j F_j$. Using this expression the probabilities read

$$p_j = \mathrm{Tr}[\varrho F_k] = \sum_k \varrho_k \mathrm{Tr}[F_j F_k] = \sum_k \varrho_k L_{jk}\,, \tag{7.7}$$

where the coefficients $L_{jk} = \mathrm{Tr}[F_j F_k]$ define a matrix L. In this setting the (inverse) problem of the state reconstruction reduces to a solution of a system of linear equations $p_j = \sum_k L_{jk}\varrho_k$, where ϱ_k are unknown. The solution exists only if the matrix L is invertible and then $\varrho_k = \sum_j L_{kj}^{-1} p_j$.

The purpose of any measurement is to provide us with an information about the state of the physical system based on results of a measurement. Our scheme of the measurement-assisted quantum processor represents a general model of a physical realization of any POVM.

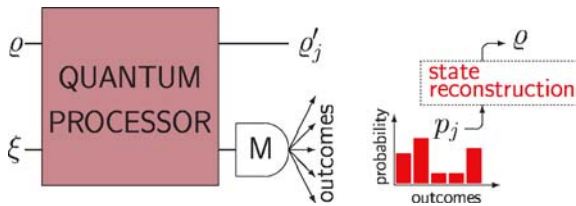

Fig. 9. Measurement-assisted quantum processors can be exploited to perform state tomography. Based on the measured probability distribution p_j one can infer the original state ϱ.

7.1.2. QID: Complete State Tomography

In what follows we shall extend the list of applications of the QID processor (see Sec. 3.3) and show how to realize a complete POVM, i.e., a complete state reconstruction. For a general program state $|\Xi\rangle = \sum_k \alpha_k |\Xi_k\rangle$ with $|\Xi_k\rangle = (\sigma_k \otimes I)|\Xi_0\rangle$ (here $|\Xi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$) the POVM consists of the following four operators

$$F_k = \sigma_k F_{0+} \sigma_k = \sigma_k A(\Xi)^\dagger A(\Xi) \sigma_k \,, \tag{7.8}$$

with $F_{0+} = \frac{1}{4} I + \frac{1}{4}[\alpha_0 \vec{\alpha^*} + \alpha_0^* \vec{\alpha} + i \vec{\alpha^*} \times \vec{\alpha}] \cdot \vec{\sigma}$ and $\vec{\alpha^*} = (\alpha_1^*, \alpha_2^*, \alpha_3^*)$, $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$.

Note that for the initial program state $|\Xi\rangle$ with $\alpha_0 = \cos\mu$, $\vec{\alpha} = i \sin\mu \frac{\vec{\mu}}{\mu}$ ($\mu = ||\vec{\mu}||$) the probabilities $p_{0+} = \mathrm{Tr} F_{0+} \varrho = 1/4$ are ϱ-independent, and a unitary operation $U_\mu = \exp(i \vec{\mu} \cdot \vec{\sigma})$ is realized. [12] The question of interest is whether an informationally complete POVM can be encoded into a program state. In fact, the problem reduces to the question of a linear independence of operators F_k for some $|\Xi\rangle$. Using the vector representation of operators, $F_k = 1/4(I + \vec{r}_k \cdot \vec{\sigma})$, one can show that the operators F_k are linearly independent only if none of the coefficients of $\vec{r}_{0+} = \alpha_0 \vec{\alpha^*} + \alpha_0^* \vec{\alpha} + i \vec{\alpha^*} \times \vec{\alpha}$ vanishes.

The elements of a POVM can be represented in the Bloch-sphere picture. This is due to the fact that operators $F_k = \frac{1}{2}\varrho_k$, and ϱ_k represent quantum states. Choosing the program state

$$|\Xi_{\mathrm{POVM}}\rangle = \frac{1}{\sqrt{2}}|\Xi_0\rangle + \frac{1}{\sqrt{6}}(|\Xi_1\rangle + |\Xi_2\rangle + |\Xi_3\rangle) \tag{7.9}$$

we obtain the informationally complete POVM with a very symmetric structure. In particular, the operators F_k are proportional to pure states associated with vertices of a tetrahedron drawn inside the Bloch sphere (see Fig. 10). These operators read

$$F_{0+} = \frac{1}{4}\left(I + \frac{1}{\sqrt{3}}[\sigma_x + \sigma_y + \sigma_z]\right); \tag{7.10}$$

$$F_{0-} = \frac{1}{4}\left(I + \frac{1}{\sqrt{3}}[-\sigma_x - \sigma_y + \sigma_z]\right); \tag{7.11}$$

$$F_{1+} = \frac{1}{4}\left(I + \frac{1}{\sqrt{3}}[\sigma_x - \sigma_y - \sigma_z]\right); \tag{7.12}$$

$$F_{1-} = \frac{1}{4}\left(I + \frac{1}{\sqrt{3}}[-\sigma_x - \sigma_y + \sigma_z]\right). \tag{7.13}$$

It is obvious that these operators are not mutually orthogonal, but $\mathrm{Tr} F_j^\dagger F_k = \frac{1}{12}\delta_{jk} + \frac{1}{4}(1 - \delta_{jk})$. Using this identity one can easily compute the

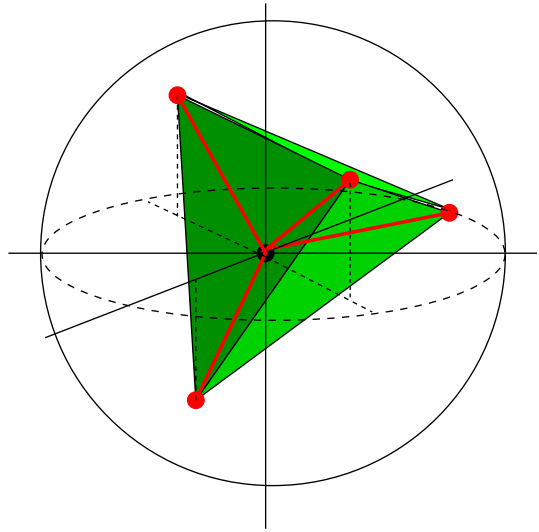

Fig. 10. The Bloch sphere can be used to illustrate any POVM that can be realized on the QID processor. Each POVM is given by four operators that determine four points in the Bloch sphere. Using this picture one can see the structure and some properties of the realized POVM. The vertices of a tetrahedron correspond to POVM elements of the symmetric informationally complete POVM associated with the program state $|\Xi\rangle = \frac{1}{\sqrt{2}}|\Xi_0\rangle + \frac{1}{\sqrt{6}}(|\Xi_1\rangle + |\Xi_2\rangle + \Xi_3\rangle)$.

relation (7.7) between the observed probability distribution and the initial data state ϱ

$$\varrho = \sum_k \left(-\frac{21}{5} p_k + \frac{9}{5} \sum_{j \neq k} p_j \right) |Q_k\rangle\langle Q_k| , \tag{7.14}$$

where we used the notation $F_k = \frac{1}{2}|Q_k\rangle\langle Q_k|$. The last equation completes the task of the state reconstruction. Because of the identity $\mathrm{Tr} F_j F_k = \mathrm{const}$ for $j \neq k$ the realized POVM $\{F_k\}$ is of a special form. It belongs to a family of the so-called symmetric informationally complete measurements (SIC POVM).[55] These measurements are of interest in several tasks of quantum information processing and possess many interesting properties. It is known (see, e.g., Ref. 55) that for qubits there essentially exist only two (up to unitaries) such measurements. Above we have shown how one of them can be performed using the QID processor.

7.1.3. von-Neumann Measurements

An important class of measurements is described by the *projector valued measures* (PVM), which under specific circumstances enable us to distinguish between orthogonal states in a single shot, i.e., no measurement statistics is required. A set of operators $\{E_k\}$ form a PVM, if $E_j = E_j^\dagger$ and $E_j E_k = E_j \delta_{jk}$, i.e., it contains mutually orthogonal projectors. The total number of (nonzero) operators $\{E_k\}$ cannot be larger than the dimension of the Hilbert space d.

Usually the *von-Neumann measurements* are understood as those that are compatible with the *projection postulate*, i.e., the result j associated with the operator $E_j = |e_j\rangle\langle e_j|$ induces the state transformation

$$\varrho \to \varrho'_j = \frac{E_j \varrho E_j}{\mathrm{Tr}\varrho E_j} = \frac{|e_j\rangle\langle e_j|\varrho|e_j\rangle\langle e_j|}{\langle e_j|\varrho|e_j\rangle} = |e_j\rangle\langle e_j| = E_j. \tag{7.15}$$

That is, the state after the measurement is described by the corresponding projector E_j.

However, each PVM can be realized in many different ways and a particular von-Neumann measurement is only a specific case. In our settings the realized POVM $\{F_k\}$ is related to the state transformation via the identity $F_k = A_k^\dagger A_k$, where $\varrho \to \varrho'_k = A_k \varrho A_k^\dagger$. The set of operators $A_k = U_k E_k$, where E_k are projectors and U_k are unitary transformations, define the same PVM given by $\{E_k\}$. In particular, $A_k^\dagger A_k = E_k U_k^\dagger U_k E_k = E_k E_k = E_k$, but the state transformation results in

$$\varrho \to \varrho'_k = U_k E_k U_k^\dagger \neq E_k. \tag{7.16}$$

Thus the final state is described by a projector, but not in accordance with the projection postulate. We refer to the PVMs that are compatible with the projection postulate as the *von-Neumann measurements*. Moreover, for simplicity we shall assume that the projectors are always one-dimensional, i.e., the PVM is associated with non-degenerate hermitian operators.

The action of the processor G implementing two von-Neumann measurements $\{E_j\}$ and $\{G_j\}$ can be written as

$$G|\psi\rangle \otimes |\Xi_E\rangle = \sum_j E_j|\psi\rangle \otimes |j\rangle; \tag{7.17}$$

$$G|\psi\rangle \otimes |\Xi_G\rangle = \sum_j G_j|\psi\rangle \otimes |j\rangle. \tag{7.18}$$

It is well known[9] that when two sets of Kraus operators are realizable by the same processor G, then the following necessary relation holds

$\sum_j E_j G_j = \langle \Xi_E | \Xi_G \rangle I$. Using this relation for the projections $E_j = |e_j\rangle\langle e_j|$, $G_j = |g_j\rangle\langle g_j|$ we obtain the identity

$$\sum_j u_{jj} |e_j\rangle\langle g_j| = kI, \tag{7.19}$$

where $u_{jj} = \langle g_j | e_j \rangle$. For general measurements, the operator on the left-hand side of the previous equation contains off-diagonal elements. In this case the corresponding program states must be orthogonal, i.e., $k = 0$. This result is similar to the one obtained by Nielsen and Chuang [7] who have studied the possibility of the realization of unitary transformations via programmable gate arrays. Nielsen and Chuang have shown that in order to perform (with certainty) two unitary transformations on a given quantum processor one needs two orthogonal program states. However, in our case we cannot be sure that given the same resources the measurement-assisted processor realizing two von-Neumann measurements does exist. Moreover, we also have to consider an option that the condition holds also for non-orthogonal program states (see the case study below). From above it follows that the realization of von-Neumann measurements on programmable processors is different from implementation unitary operations on programmable processors. The reason is that for implementation of von-Neumann measurements program states might not satisfy the criterion in Eq. (7.19).

Orthogonal program states

In order to realize a measurement described by PVM (either a von-Neumann measurement or a general PVM measurement) on a d-dimensional data register the program space must be at least d dimensional. Let us start with the assumption that the Hilbert space of the program register is d dimensional and the program states are orthogonal. Our task is to analyze the possibility to perform d different (non-degenerate) von-Neumann measurements M_α determined by a set of operators $E_k^\alpha = |\alpha_k\rangle\langle\alpha_k|$ ($E_k^\alpha E_j^\alpha = \delta_{kj} E_k^\alpha$ and $\sum_k E_k^\alpha = I$ for all α). Let $|\alpha\rangle$ denote the associated program states and $\langle\alpha|\beta\rangle = \delta_{\alpha\beta}$. It is easy to see that for general measurements the resulting operator

$$G = \sum_{k,\alpha} E_k^\alpha \otimes |k\rangle\langle\alpha| \tag{7.20}$$

is not unitary. In particular, $G^\dagger G = \sum E_k^\beta E_k^\alpha \otimes |\beta\rangle\langle\alpha| \neq I$. The equality would require that the identity $\sum_k E_k^\alpha E_k^\beta = \delta_{\alpha\beta} I_d$ holds. Therefore, we conclude that neither orthogonal states do guarantee the existence of a

programmable processor that performs desired set of von-Neumann measurements. This result makes the programming of unitaries and programming of von-Neumann measurements different.

For instance, let us consider a two-dimensional program register and let us denote $E^0_{0,1} = E_{0,1}$ and $E^1_{0,1} = G_{0,1}$. Then the above condition reads $E_0 G_0 = E_1 G_1 = 0$. Using the definition $E_k = |e_k\rangle\langle e_k|$ and $G_k = |g_k\rangle\langle g_k|$ we obtain the orthogonality conditions $\langle e_0|g_0\rangle = \langle e_1|g_1\rangle = 0$. Consequently, because in the two-dimensional case the orthogonal state is unique, we obtain $|g_0\rangle = |e_1\rangle$ and $|g_1\rangle = |e_0\rangle$, i.e., the measurements are the same. Similarly one can show that even for qutrit ($d_p = d = 3$) one can perform only one von-Neumann measurement, too. In particular, $\langle e_0|g_0\rangle = 0$ implies $|g_0\rangle = a_0|e_1\rangle + b_0|e_2\rangle$, $|g_1\rangle = a_1|e_0\rangle + b_1|e_2\rangle$ and $|g_2\rangle = a_2|e_0\rangle + b_2|e_1\rangle$. Orthogonality of $|g_j\rangle$ results in a set of equations $a_0 b_2 = 0$, $b_0 b_1 = 0$, $a_1 a_2 = 0$ with the solution that set of vectors $\{|g_j\rangle\}$ is just a permutation of the set $\{|e_j\rangle\}$. However, this solution does not correspond to a realization of two non-commutative measurements. To perform two such measurements one need an extra dimension, i.e., for $d_p = d = 4$ we can realize two von-Neumann measurements. An addition of new dimension enables us to perform one more non-commuting measurement, [56] i.e., with d orthogonal states one can implement at most $N = d - 2$ non-commuting von-Neumann measurements. See Table 1. for the properties that the corresponding eigenvectors have to satisfy.

In order to implement a set of von-Neumann measurements on a qudit (with d-dimensional Hilbert space) one has to utilize a program register with the dimension of the Hilbert space such that $\dim \mathcal{H}_p = d_p > d$. In general, in this case we work with d_p outcomes and d_p projective operators Q_k that define the realized measurement. However, each PVM consists of maximally d projectors. Therefore, $d_p - d$ of the induced operators E_k should represent the *zero operator*. It means that we are realizing the

Table 1. The measurements $M_1, M_2, \ldots, M_N$ are realizable by a d dimensional program register only if all vectors in the rows are mutually orthogonal. Moreover, no two columns can be related by a permutation. The orthogonality of the vectors in columns is ensured by the fact that they form a PVM. It turns out that the number of realizable measurements equals to at most $N - 2$, i.e., even with qutrit one cannot encode more than a single von-Neumann measurement. Moreover, the measurements that can be performed are not arbitrary.

Measurement	M_1	M_2	$\ldots$	M_N
Result 1	$\|\alpha_1\rangle$	$\|\beta_1\rangle$	$\ldots$	$\|\omega_1\rangle$
Result 2	$\|\alpha_2\rangle$	$\|\beta_2\rangle$	$\ldots$	$\|\omega_2\rangle$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Result d	$\|\alpha_d\rangle$	$\|\beta_d\rangle$	$\ldots$	$\|\omega_d\rangle$

von-Neumann measurement such that some of the outcomes do not occur, i.e., the probability of these outcomes is equal to zero for all data states. However, there is one more option that the set of operators $\{E_k\}$ (corresponding to the outcomes $k = 1, \ldots, d_p$) contains exactly only d different operators (projectors). This means that more outcomes specify the same projection and define a single result of the realized von-Neumann measurement.

We can utilize the so-called "zero" operators to formulate a general approach how to implement any set of arbitrary von-Neumann measurements. Let us consider N von-Neumann measurements M_α $(\alpha = 1 \ldots N)$ given by non-zero operators $\{E_k^\alpha\}$ (number of k equals to d). We can define new sets of d_p operators $\{\tilde{E}_k^\alpha\}$ by adding to $\{E_k^\alpha\}$ zero operators so that the condition $\sum_k \tilde{E}_k^\alpha \tilde{E}_k^\beta = \delta_{\alpha\beta} I$ holds. Using this approach we find that any collection of N von-Neumann measurements can be realized on a single quantum processor given by Eq. (7.20) with (maximally) $N \cdot d$ dimensional program space.

Let us summarize our results in the following propositions:

Proposition 1. Using $d_p = d$ dimensional program space and orthogonal program states allows us to encode maximally $N = d - 2$ specific (non-commuting) von-Neumann measurements on a qudit (see Table 1).

Proposition 2. Let $M_1, \ldots, M_N$ be N (non-commuting) von-Neumann measurements on a qudit. Then it is sufficient to use $N \cdot d$-dimensional program space to encode these measurements into orthogonal program states.

Case Study: Projective Measurements on a Qubit

Let us consider two von-Neumann measurements $M = \{E_0, E_1\}$ and $N = \{G_0, G_1\}$ on a qubit. Firstly, we will assume a three-dimensional Hilbert space of a program register. We define measurements $M_1 = \{E_0, E_1, 0\}$ and $M_2 = \{0, G_1, G_2\}$, respectively. It is easy to see that neither of these two sets of operators do satisfy the condition $0 = \sum_k E_k G_k = 0E_0 + E_1 G_1 + 0G_2 = E_1 G_1$. The equality holds only if $E_1 G_1 = 0$, i.e., $E_1 = |\psi\rangle\langle\psi|$ and $G_2 = |\psi_\perp\rangle\langle\psi_\perp|$, but this implies that the two measurements are the same. Consequently, the dimension of the program space has to be increased in order to encode into a program register two projective measurements on a qubit. Therefore, let us consider a four-dimensional Hilbert space of the program register. In this case we have $M_1 = \{E_0, E_1, 0, 0\}$, $M_2 = \{0, 0, G_0, G_1\}$ and the condition holds for all possible measurements M_1, M_2. We conclude that in order to implement N von-Neumann measurements (by encoding into orthogonal states) on a qubit

a $2N$-dimensional program space is required. Let us note that for qudits this is only the sufficient condition and for specific collections of measurements we can do better.

In what follows we shall show a way how to realize three different von-Neumann measurements on a qubit by using only four-dimensional program space. To achieve this goal we will use *non-orthogonal* program states. We will show that in special cases the condition of orthogonality [given by Eq. (7.19)] can be relaxed. The program space of the QID processor given by Eq. (3.36) consists of two qubits. Using the conclusion of the previous paragraph we see that QID allows us to perform two von-Neumann measurements. It is easy to see that the operators $A_k = \sigma_k A(\Xi)\sigma_k$ with $A(\Xi) = \frac{1}{2}\sum_j \alpha_j \sigma_j$ are not projectors. Consequently, the projective measurement cannot be realized in the same way as described above. However, the QID-processor can still be exploited to perform a von-Neumann measurement.

Using the program state $|\Xi\rangle = \frac{1}{\sqrt{2}}(|\Xi_0\rangle + |\Xi_1\rangle)$ the operator $A = \frac{1}{2\sqrt{2}}[I + \sigma_x]$ (i.e., $F_0 = A^\dagger A = \frac{1}{2}P_+$, where $P_+ = \frac{1}{2}[I + \sigma_x]$) is a projection onto the vector $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + |1\rangle$. It is obvious that $F_1 = \sigma_x F_0 \sigma_x = F_0$ and $F_2 = F_3 = \frac{1}{2}P_-$, where $P_- = \frac{1}{2}[I - \sigma_x]$. It turns out that we have realized the PVM described by $P_\pm$, i.e., the eigenvectors of the σ_x measurement. The state transformation reads $\varrho \to \varrho'_k = P_\pm$ (if $p_k \neq 0$), respectively. It follows that the realization of the measurement of σ_x is in accordance with the projection postulate. In the same way we can realize σ_y and σ_z measurement (in these cases different results must be paired). Basically, this corresponds to a choice of different two-valued measurements, but in reality we perform only a single four-valued measurement. As a result we find that on the QID we can realize three different von-Neumann measurements. Note that we have used only two qubits as the program register. Moreover, the associated program states $|\Xi_{\sigma_j}\rangle = \frac{1}{\sqrt{2}}[|\Xi_0\rangle + |\Xi_j\rangle]$ are not mutually orthogonal, but $\langle\Xi_{\sigma_j}|\Xi_{\sigma_k}\rangle = \frac{1}{2}$ (for $j \neq k$) and Eq. (7.19) holds. Namely, for the measurements of $\sigma_x \leftrightarrow \{P_\pm\}$ and $\sigma_z \leftrightarrow \{P_0 = |0\rangle\langle 0|, P_1 = |1\rangle\langle 1|\}$ the condition (7.19) reads $\frac{1}{2}[P_+P_0 + P_+P_1 + P_-P_1 + P_-P_0] = \frac{1}{2}I$.

7.1.4. Projection Valued Measures

If we relax the projection postulate more PVMs can be realized on a single processor. Let us consider that the dimension of the program space equals d and $|\alpha\rangle$ is the state that encodes the PVM given by a set $\{E_k^\alpha\}$. The action of G can be written as

$$G|\psi\rangle \otimes |\alpha\rangle = \sum_k U_k^\alpha E_k^\alpha |\psi\rangle \otimes |k\rangle \tag{7.21}$$

and the condition $\sum_k E_k^\alpha U_k^{\alpha\dagger} U_k^\beta E_k^\beta = \delta_{\alpha\beta} I$ must hold. Let us consider two PVMs on a qubit $\{E_0 = |0\rangle\langle 0|, E_1 = |1\rangle\langle 1|\}$ and $\{G_0 = |\phi\rangle\langle\phi|, G_1 = |\phi_\perp\rangle\langle\phi_\perp|\}$. Define a unitary map U such that $|\phi\rangle \to |1\rangle$ and $|\phi_\perp\rangle \to |0\rangle$. Using this map we can define a processor by following equations

$$\begin{aligned} G|\psi\rangle \otimes |\Xi_E\rangle &= E_0|\psi\rangle \otimes |0\rangle + E_1|\psi\rangle \otimes |1\rangle; \\ G|\psi\rangle \otimes |\Xi_G\rangle &= \tilde{G}_0|\psi\rangle \otimes |0\rangle + \tilde{G}_1|\psi\rangle \otimes |1\rangle, \end{aligned} \tag{7.22}$$

where $\tilde{G}_0 = UG_0 = |1\rangle\langle\phi|$, $\tilde{G}_1 = UG_1 = |0\rangle\langle\phi_\perp|$ and $\langle \Xi_E | \Xi_G \rangle = 0$. Direct calculation shows that $E_0\tilde{G}_0 + E_1\tilde{G}_1 = |0\rangle\langle 0|1\rangle\langle\phi| + |1\rangle\langle 1|0\rangle\langle\phi_\perp| = 0$, i.e., G is unitary. From here it follows that if one does not require the validity of the projection postulate, then any two PVMs can be performed on a processor with two-dimensional program space.

This result holds in general. Let us consider a set of d PVMs $\{E_k^\alpha\}$ on a qudit. There always exist unitary transformations U^α such that operators $\tilde{E}_k^\alpha = U^\alpha E_k^\alpha$ satisfy the condition $\sum_k \tilde{E}_k^{\alpha\dagger} \tilde{E}_k^\beta = \delta_{\alpha\beta} I$. Without the loss of generality we can consider that the measurement M_0 is given by projectors $|0\rangle\langle 0|, \ldots, |d-1\rangle\langle d-1|$ and M_α by $|\phi_0^\alpha\rangle\langle\phi_0^\alpha|, \ldots, |\phi_{d-1}^\alpha\rangle\langle\phi_{d-1}^\alpha|$ (see Table 2).

Proposition 3. A collection of arbitrary (non-commuting) N projection-valued measures can be realized on quantum processor with N-dimensional program space.

Table 2. The realization of an arbitrary collection of d PVMs $M_1, \ldots, M_d$ on a qudit. The operators $\tilde{E}_k^\alpha$ correspond to an example of the choice of unitary transformations U^α. In particular, each U^α transforms the basis $\{|\phi_j^\alpha\rangle\}$ into some permutation of the basis $\{|j\rangle\}$. The permutation is different for each α.

$M_1 \leftrightarrow \tilde{E}_k^1$	$M_2 \leftrightarrow \tilde{E}_k^2$	...	$M_d \leftrightarrow \tilde{E}_k^d$
$\|0\rangle\langle 0\|$	$\|1\rangle\langle\phi_0^2\|$	...	$\|d-1\rangle\langle\phi_0^d\|$
$\|1\rangle\langle 1\|$	$\|2\rangle\langle\phi_1^2\|$	...	$\|0\rangle\langle\phi_1^d\|$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\|d-1\rangle\langle d-1\|$	$\|0\rangle\langle\phi_{d-1}^2\|Z$	...	$\|d-2\rangle\langle\phi_{d-1}^d\|$

7.1.5. Programming Unitaries vs PVMs

As an alternative to the scenario presented in the previous section one can consider the following strategy how to realize a measurement on the programmable quantum processor. Programmable processors are designed to perform unitary operations. Since different projection valued measures are always related by some fixed unitary transformation, it is possible to exploit the existing processor to rotate the input data state by a suitable transformation. After this transformation is implemented the *fixed* von-Neumann measurement of the data register is performed. In particular, let us consider that the processor G implements the transformation $|\psi\rangle \rightarrow U|\psi\rangle$ and the fixed measurement of the data register is described by set of projectors $\{E_k\}$. Using such a processor the measured probabilities read $p_k = \langle\psi|U^\dagger E_k U|\psi\rangle = \langle\psi|F_k|\psi\rangle$, where operators $F_k = U^\dagger E_k U$ describe the realized PVM. However, the output state $|\psi'_k\rangle$ is described by the corresponding projection E_k. To obtain the state transformation that is in accordance with the projection postulate one has to apply the same unitary transformation U once more, i.e., we use the same processor twice.

From here it follows that the implementation of a von-Neumann measurement is related to a repeated usage of the processor realizing the given unitary operation. In particular, to realize N von-Neuman measurements we have to use twice the processor realizing N unitary transformations, i.e., the program space is composed of two N-dimensional systems (unitary operators are encoded in orthogonal states). As a result we find that the dimension of the program space equals to N^2. In the limit of large number of measurements this N^2 is larger than Nd that quantifies the number of orthogonal program states from the Proposition 2. In fact, whenever the number of measurements is larger than the dimension of the object, the usage of a quantum processor realizing unitary transformations is less efficient.

Let us note that with this realization of measurements we do not have to consider the compatibility with the projection postulate, providing that the measurement is not performed in a non-demolition way. However, non-demolition measurements require additional systems and therefore the model would correspond again to some measurement-assisted quantum processor. If one does not care about a particular realization of the PVM, then the number of realizable PVMs N equals to the dimension of the number program states encoding the corresponding unitary transformations. This is exactly the content of the Proposition 3.

Comments

Above we have studied how POVMs can be physically realized using the so-called measurement-assisted quantum processors. In particular,

we have analyzed how to perform a complete state reconstruction and von-Neumann measurements. As a result we have found that an arbitrary collection of von-Neumann measurements cannot be realized on a single programmable quantum processor of finite dimension. We have shown how to use the QID processor to perform the state reconstruction.

The number of implementable von-Neumann measurements is limited by the dimension of the program register. Our main result is that with a program register containing Nd orthogonal states one can certainly find a processor which performs arbitrary N von-Neumann measurements. In principle, one can do much better than this. We have shown that non-orthogonal program states can be used very efficiently. This makes the programmability of unitary transformations and von-Neumann measurements different. In particular, the QID processor can be exploited to perform three von-Neumann measurements by using three non-orthogonal states of only two qubits of the program register. Using $d_p = d$ dimensional program space one can encode maximally $N = d - 2$ von-Neumann qudit measurements into orthogonal program states. (for a qubit we have $N = 1$).

Relaxing the condition of compatibility with the projection postulate the processor allows us to realize any collection of N PVMs by using only $d_p = N$ dimensional program space. An open question is whether we can perform more PVMs or not. The two tasks can be performed by programmable processors: the realization of von-Neumann measurements and the application of unitary transformations on the data register. This two applications are different. According to Nielsen and Chuang, [7] any collection of N unitary transformations requires N dimensional program space. For N von-Neumann measurements the upper bound reads $d_p = Nd$ and any improvement strongly depends on the specific set of these measurements.

7.2. Measurement of Husimi Function with Programmable Processors

In what follows we will show how quantum filtering of the original (input) data register can be realized and how propensities (e.g., a Husimi function) of the input register can be easily measured using the programmable processor as represented by the QID.

7.2.1. Quantum Propensities

According to Wódkiewicz,[57] propensity means the tendency (or probability) of a measured object to take up certain states prescribed by a measuring device. Let the measuring device—the so-called quantum ruler—be in a pure state $|\Phi\rangle$. The quantum-ruler state can be "shifted" by

an action of some generalized displacement operator $D(g)$, where g is an element of a group G. If the measured system is in a pure state $|\Psi\rangle$, then its probability to be in the ruler state shifted by g (i.e., the propensity) is

$$P_{\Phi,\Psi}(g) = |\langle\Psi|D(g)|\Phi\rangle|^2, \tag{7.23}$$

whereas if the system is in a mixed state described by the density operator ρ, the propensity is

$$P_{\Phi,\rho}(g) = \mathrm{Tr}\left(\rho D(g)|\Phi\rangle\langle\Phi|D^{+}(g)\right). \tag{7.24}$$

In our case, that of a finite dimensional Hilbert space, the group G will be formed by discrete translations on a torus: if $g_1 \equiv (n_1, m_1)$ and $g_2 \equiv (n_2, m_2)$ are elements of G, then their group product is $g_1 g_2 \equiv ((n_1 + n_2)\mathrm{mod}N, (m_1 + m_2)\mathrm{mod}N)$. The corresponding displacement operator is then given by the expression $R_x(n)R_p(m)$. We see that while the displacement is not a representation of the group G in the Hilbert space under consideration, nevertheless it is representation of this group in a ray space, which enables us to define the propensity uniquely. For a pure state $|\Psi\rangle$ we can write the propensity in the form (see Ref. 36):

$$P_{\Phi,\Psi}(n, m) = |\langle\Psi|R_x(n)R_p(m)|\Phi\rangle|^2. \tag{7.25}$$

In the case of a statistical mixture described by the density operator ρ the corresponding propensity reads

$$P_{\Phi,\rho}(n, m) = \mathrm{Tr}\left[\rho R_x(n)R_p(m)|\Phi\rangle\langle\Phi|R_p^{\dagger}(n)R_x^{\dagger}(m)\right]. \tag{7.26}$$

7.2.2. Propensities and POVM Measurements

The propensities as defined above are in fact results of so-called generalized (POVM) measurements (e.g., see Ref. 2). To see this let us recall that

$$F_{mn} = R_x(n)R_p(m)|\Phi\rangle\langle\Phi|R_p^{\dagger}(m)R_x^{\dagger}(n) \tag{7.27}$$

where $|\Phi\rangle$ is a ruler state are *positive* operators and they fulfill the condition

$$\sum_{mn} F_{mn} = NI\,. \tag{7.28}$$

So the operators F_{mn} (or more specifically the operators $f_{mn} = F_{mn}/N$) form a complete set that can be used for a complete measurement of the state of a qudit. We note that other operators of the form Eq. 7.27, e.g.,

$$F_{mn} = R_x(m) R_p^\dagger(n) \rho R_p(n) R_x^\dagger(m) \tag{7.29}$$

also realize a POVM measurement.

7.2.3. *Q-function in Discrete Phase Space*

In an analogy with a continuous (q, p) phase space, where the Q-function (Husimi function) is defined as the propensity of a state to be in the vacuum state, we define the discrete Q-function as the propensity (7.23)

$$Q(n, m) \equiv P_{\Phi, \rho}(n, m), \tag{7.30}$$

with the quantum ruler being in a "vacuum" state. The problem is how to define a vacuum state corresponding to a finite-dimensional Hilbert space.

Before specifying the ruler state, we will mention several properties of discrete Q-functions. If we assume that the ruler state $|\Phi\rangle$ is chosen (i.e., the vacuum state is specified) then the Q-function has the following properties:

(i) it is uniquely defined;
(ii) it is non-negative;
(iii) it is normalized to N

$$\sum_{n,m} Q(n, m) = N; \tag{7.31}$$

(iv) for *properly* chosen ruler states $|\Phi\rangle$ the information about a system state can be completely reconstructed from the corresponding Q-function.

7.2.4. *Ruler State*

In analogy with the continuous limit, where the ruler state associated with a Husimi function is the ground (vacuum) state of the harmonic oscillator, let us consider following requirements on the ruler state: (i) it should be in some sense centered at origin of phase space [i.e., the point $(0, 0)$], (ii) it should be "symmetric" with regards to the quantities X and P, i.e., its wave function should have similar form in both representations (perhaps up to scalings), and (iii) it should be in some sense a minimum

uncertainty state, which means that in the phase space it should be represented by a peak which is as narrow as possible. As shown in Ref. 31 all the above properties are fulfilled by the ground state of the Hamiltonian

$$H_0 = -\cos\left(\frac{2\pi}{N}X\right) - \cos\left(\frac{2\pi}{N}P\right). \tag{7.32}$$

We will use this ground state as the ruler state in our forthcoming considerations.

7.2.5. Measurement of Husimi Function via QID

Let us now study the action of the quantum information distributor when the two ancillary qudits are prepared in a superposition state

$$|\Theta\rangle_{23} = (\alpha|\Xi_{00}\rangle_{23} + \beta|x_m\rangle_2|p_n\rangle_3), \tag{7.33}$$

with the two real amplitudes α and β satisfying the normalization condition

$$\alpha^2 + \beta^2 + \frac{2\alpha\beta}{N}\cos\left(\frac{2\pi}{N}nm\right) = 1. \tag{7.34}$$

With this program state the QID acts on the input data qudit $|\Psi\rangle_1 = \sum_k c_k|x_k\rangle$ so that at the output the three qudits are in the following states:

$$\rho_1 = (1-\beta^2)\rho + \frac{\beta^2}{N}I; \tag{7.35}$$

$$\rho_2 = (1-\alpha^2)R_x(m)R_p^\dagger(n)\rho R_p(n)R_x^\dagger(m) + \frac{\alpha^2}{N}I; \tag{7.36}$$

$$\rho_3 = (1-\alpha^2-\beta^2)R_x(m)R_p(n)\rho^{\mathrm{T}}R_p^\dagger(n)R_x^\dagger(m) + \frac{\alpha^2+\beta^2}{N}I, \tag{7.37}$$

where $\rho = |\Psi\rangle\langle\Psi|$ and ρ^{T} is the transpose of the density operator $\rho = \sum_{k,k'} c_k c_{k'}^* |x_k\rangle\langle x_{k'}|$. That is, in the basis $|x_k\rangle$ the transposed density operator reads $\rho^{\mathrm{T}} = \sum_{k,k'} c_k^* c_{k'} |x_k\rangle\langle x_{k'}|$.

The action of the QID discussed earlier, allows us to reconstruct partially the state of the measured system without a total "destruction" of the state of the data register. From our discussion in Sec. 3.3 that the entangled component of the program register (represented by the state $|\Xi_{00}\rangle_{23}$) dictates how "much" of the original information encoded in the qudit 1

is transferred from the data register to the program register at the output of the QID. For instance, if the amplitude α is equal to unity (i.e., $\beta = 0$) then the data register is not perturbed at all, and no information is transferred. On the other hand, for $\alpha < 1$ some of the information from the data is transferred to the program at the expense of noise introduced into the data register. The trade-off between the information transfer and the noise introduced into the data register is nicely seen from Eq. 5.3. The amount of noise that is transferred into the first (data) qudit is dictated by the amplitude β that weights the factorizable contribution to the program state, i.e. $|x_n\rangle_2|p_m\rangle_3$. Moreover, this specific state also determines operations (rotations) that are performed on program qudits.

In order to illustrate the action of the QID we plot in Fig. 11 Q-functions of an input qudit that is initially prepared in the ground state of the Hamiltonian (7.32), as well as the three output qudits. The ruler state is chosen to be again the ground state of the Hamiltonian (7.32). The Husimi functions do correspond to the situation when a POVM measurement is performed on the density operator ρ_j $(j = 1, 2, 3)$ given by Eqs. 7.35–7.37, respectively.

It is obvious from the expression (7.36) that if the von Neumann measurement using the projector $|\Phi\rangle_2\langle\Phi|$ (i.e., projecting on the ruler state) on the qudit 2 is performed then this measurement results in a reconstruction of the Husimi function of the original data state affected by the amount of noise determined by the particular value of α. In other words, this projective measurement will result in the reconstruction of the Husimi function of the operator $\rho_2^{(\mathrm{out})} = (1-\alpha^2)\rho + \frac{\alpha^2}{N} I$. Certainly, the state of the data register is then affected not only by the action of the QID but also by the effect of the projective measurement performed on the second qudit.

To understand the role of the projective measurement performed on the program register on the state of the data register at the output of the QID, let us consider the following. We will study the action the quantum information distributor when the two ancillary qudits are prepared in a superposition state given by Eq. 7.33. With this program state the QID acts on the input data qudit $|\Psi\rangle_1 = \sum_k c_k|x_k\rangle$ so that at the output the three qudits are in the state:

$$\begin{aligned} |\Omega^{(\mathrm{out})}\rangle_{123} &= P_{123}|\Psi\rangle_1\left[\alpha|\Xi_{00}\rangle_{23} + \beta|x_m\rangle_2|p_n\rangle_3\right] \\ &= \alpha|\Psi\rangle_1|\Xi_{00}\rangle_{23} + \beta\left[R_x(m)R_p^{\dagger}(n)|\Psi\rangle\right]_2|\Xi_{nm}\rangle_{31}\,. \end{aligned} \tag{7.38}$$

Then we will assume that both program qudits are measured projectively. The qudit 2 is projected in the ruler state $|\Phi\rangle_2 = \sum_k f_k|x_k\rangle_2$ while the qudit

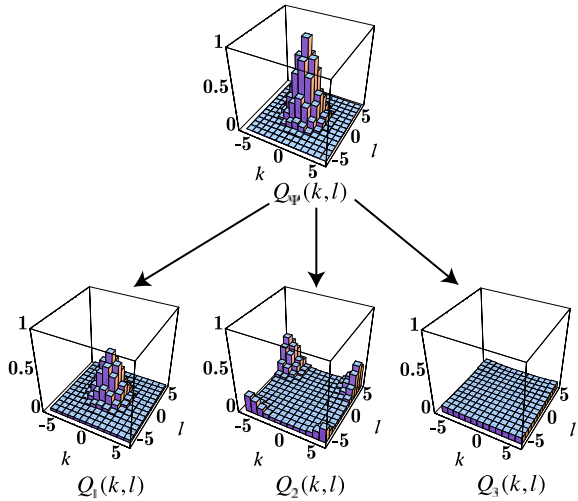

Fig. 11. Husimi functions of the input state of the data qudit and the output qudits. The input data qudit is initially prepared in the ground state of the Hamiltonian (7.32) while the auxiliary system (ancilla) is initially prepared in the state $|\Theta\rangle = 0.75|\Xi_{00}\rangle - 0.64|x_7\rangle|p_5\rangle$. The top graph, labeled $Q_\Psi(k,l)$, represents the Husimi function of the initial state of the data qudit. The three graphs, labeled $Q_1(k,l)$, $Q_2(k,l)$ and $Q_3(k,l)$, represent the Husimi functions of reduced states ρ_1, ρ_2 and ρ_3 of the composite system that are given by Eqs. (7.35–7.37), respectively.

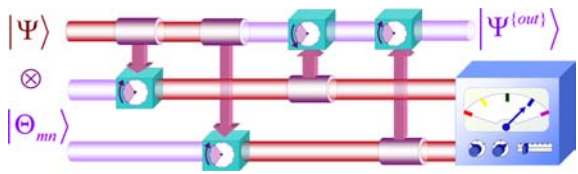

Fig. 12. Logical network for the quantum information distributor with a projective measurement performed on the program register.

3 is projected on the transposed ruler state $|\Phi^{\mathrm{T}}\rangle_3 := \sum_k f_k^* |x_k\rangle_3$. Schematically this situation is depicted in Fig. 12.

The data qudit after the action of the QID and this projective measurement reads

$$|\Psi^{(\mathrm{out})}\rangle_1 \simeq_2 \langle\Phi|_3\langle\Phi^{\mathrm{T}}|\Omega^{(\mathrm{out})}\rangle_{123}$$
$$= \frac{\alpha}{\sqrt{N}}|\Psi\rangle_1 + \frac{\beta\langle\Phi|R_x(m)R_p^{\dagger}(n)|\Psi\rangle}{\sqrt{N}} R_x^{\dagger}(m)R_p(n)|\Phi\rangle_1. \qquad (7.39)$$

This means that by acquiring knowledge of a particular value of the Husimi functions of the second and the third qudits, the data qudit "collapses" into the state (7.39). The disturbance of the original data state depends on the value of α, the particular point (m, n) at which the Husimi functions of the program qudits are measured and the specific choice of the ruler state.

Comments

From the above discussion it follows that it is possible to use the QID to measure the discrete Q function of an arbitrary qudit that serves as the input data of the programmable processor. This is equivalent to realizing a class of POVM operators. Another possibility, is to split the input into two parts, to find the Q function of one part and retain the other part. There is a trade-off involved: the more information that is retained, the more smeared is the Q function, and the better the Q function, the more distorted is the information in the retained qudit. Thus, the QID provides us with a very flexible programmable quantum information processing device, which has a number of useful applications.

7.3. Programmable State Discriminators

Now we focus our attention on another class of programmable devices. We will analyze whether it is possible to construct a universal (multi-purpose) quantum measurement device ("quantum multi-meter"). That is, an apparatus that could perform a specific class of generalized measurements (POVM) in such a way that each member of this class could be selected by a particular quantum state of a "program register." The key property of this approach is a possibility to control the choice of the measurement (e.g., the measurement basis in case of a projective measurement) by a (in principle, unknown) quantum state of the program register. This state can be determined, for instance, as a result of some quantum-information process.

As discussed above, the generalized POVM measurement[13–15] is defined by the fact that the probability of each of its result (the number of results may be, in general, larger than the dimension of the Hilbert space of the measured system) is given by the expression $p_\mu = \mathrm{Tr}_{\mathrm{S}}\left(\mathsf{F}_\mu \rho_{\mathrm{S}}\right)$, where ρ_{S} is the state of the system and F_μ are *positive* operators that constitute

the decomposition of the identity operator ($\sum_\mu \mathsf{F}_\mu = I$). Each POVM can be implemented using an ancillary quantum system in a specific state and realizing a projective von Neumann measurement on the composite system.[14] In other words, if one has an "input" (measured) state ρ_S in the Hilbert space $\mathcal{H}_S$ it is always possible to find some state ρ_A in a space $\mathcal{H}_A$ and a set of orthogonal projectors $\{\mathsf{E}_\mu\}$ acting on $\mathcal{H}_S \otimes \mathcal{H}_A$ ($\sum_\mu \mathsf{E}_\mu = I$) such that

$$\mathsf{F}_\mu = \mathrm{Tr}_A \left(\mathsf{E}_\mu \rho_A\right) \tag{7.40}$$

are positive operators as discussed above.

In general, we can assume, that the initial state of the ancilla can be prepared with an arbitrary precision. The ancilla can be considered as a part of the "program register." Further, we note that the general projection measurement on the composite system can be represented by a unitary transformation on the composite system followed by a fixed projection measurement (e.g., independent projective measurements on individual qubits). Therefore the problem of designing the programmable quantum multi-meter reduces to the question whether an arbitrary unitary operation (on the Hilbert space with a given dimension) can be encoded in some quantum state of a program register of a finite dimension. It has been proved by Nielsen and Chuang[7] that any two inequivalent operations require orthogonal program states. Thus the number of encoded operations cannot be higher than the dimension of the Hilbert space of the program register.

In general, we can describe a quantum multi-meter as a (fixed) unitary operation acting on the measured system (or a "data register") and an ancillary system ("program register") together and a (fixed) projective measurement realized afterwards on the same composite system. Clearly, such a device can perform only a restricted set of POVM's. One can, therefore, ask what is the optimal unitary transformation that enables us to implement "the largest set of POVM's" (in comparison with the set of POVM's that would be obtainable when we allowed any unitary transformation on the same Hilbert space). One can also ask what unitary transformation can help to approximate all the POVM's (generated by an arbitrary unitary transformation) with the highest precision (fidelity) on average. Clearly, the last task requires definition of the distance measure between two POVM's. This is an interesting problem *per se*, however, it goes far beyond the scope of our considerations here. Both optimization problems mentioned above are rather non-trivial. Moreover, the introduced scheme is perhaps too general from a practical point of view. Therefore, below we will concentrate our attention on a more specific case: On the problem of state discrimination.

We stress once again that a quantum multi-meter as discussed below is a device which in contrast to its classical counterpart is controlled (switched, programmed) by quantum states of a program register that are allowed to be mutually non-orthogonal.

7.3.1. Discrimination of Quantum States

Let us study a particular example of a "quantum multi-meter" serving for a programmable unambiguous state discrimination. So, it is in place to say a few words about quantum state discrimination now.

A general *unknown* quantum state cannot be determined completely by a measurement performed on a single copy of the system. But the situation is different if *a priori* knowledge is available[13–15]—e.g., if one works only with states from a certain discrete set. Even quantum states that are mutually non-orthogonal can be distinguished with a certain probability provided they are linearly independent (for a review see Ref. 58). There are, in fact, two different optimal strategies[59]: First, the strategy that determines the state with the minimum probability for the error[13,14] and, second, unambiguous or error-free discrimination (the measurement result never wrongly identify a state) that allows the possibility of an inconclusive result (with a minimal probability in the optimal case).[60–64] We will concentrate our attention to the unambiguous state discrimination. It has been first investigated by Ivanovic[60] for the case of two equally probable non-orthogonal states. Peres[2] solved the problem of discrimination of two states in a formulation with POVM measurement. Later Jaeger and Shimony[63] extended the solution to arbitrary *a priori* probabilities. Chefles and Barnett[64] have generalized Peres's solution to an arbitrary number of equally probable states which are related by a symmetry transformation. Unambiguous state discrimination were already realized experimentally. The first experiment, designed for the discrimination of two linearly polarized states of light, were done by Huttner *et al.*[65] There are also some newer proposals of optical implementations.[66] The interest in the quantum state discrimination is not only "academic"—unambiguous state discrimination can be used, e.g., as an efficient attack in quantum cryptography.[67]

7.3.2. "Universal" Discriminator

Let us suppose that we want to discriminate unambiguously between two known non-orthogonal states. However, we would like to have a possibility to "switch" the apparatus in order to be able to work with several different pairs of states.

Let us have two (non-orthogonal) input states of a qubit. We can always choose such a basis that they read $\alpha_0|0_D\rangle \pm \beta_0|1_D\rangle$ with $\alpha_0 = \cos(\varphi_0/2)$ and $\beta_0 = \sin(\varphi_0/2)$; the value of φ_0 can be from 0 to $\pi/2$ (φ_0 is the angle between the two states). Let us have one additional ancillary qubit, initially in a state $|0_A\rangle$. On both the "data" and the ancilla we apply the following unitary transformation $\mathcal{U}_{DA}$:

$$\begin{aligned}
&|0_D0_A\rangle \rightarrow \cos\theta\,|0_D0_A\rangle + \sin\theta\,|0_D1_A\rangle,\\
&|1_D0_A\rangle \rightarrow |1_D0_A\rangle,\\
&|0_D1_A\rangle \rightarrow -\sin\theta\,|0_D0_A\rangle + \cos\theta\,|0_D1_A\rangle,\\
&|1_D1_A\rangle \rightarrow |1_D1_A\rangle,
\end{aligned} \tag{7.41}$$

where $\cos\theta = \tan(\varphi_0/2)$. If we then make a von Neumann measurement consisting of the projectors $\mathsf{P}_+ = |+\rangle\langle+|$, $\mathsf{P}_- = |-\rangle\langle-|$, and $\mathsf{P}_0 = I - \mathsf{P}_+ - \mathsf{P}_-$, where

$$|\pm\rangle = (|0_D0_A\rangle \pm |1_D0_A\rangle)/\sqrt{2}, \tag{7.42}$$

we can unambiguously determine the input state (with a certain probability of the success). This measurement is optimal in the sense that the probability of inconclusive result is the lowest possible (and it is the same for both states). The probability of the successful discrimination is $2\sin^2(\varphi_0/2)$.[62]

Let us suppose now the set of pairs

$$\begin{aligned}
|\psi_1\rangle &= \alpha\,|0_D\rangle + \beta\,|1_D\rangle,\\
|\psi_2\rangle &= \alpha\,|0_D\rangle - \beta\,|1_D\rangle,
\end{aligned} \tag{7.43}$$

where $\alpha = \cos(\varphi/2)$ and $\beta = \sin(\varphi/2)$, for all φ from the interval $(0, \pi)$. That is, we consider all pairs of states that lie on a real plane and that are located symmetrically around the state $|0_D\rangle$; see Fig. 13. Further, let us suppose that the ancillary qubit is allowed to be in an arbitrary pure state

$$|\Xi\rangle_A = a|0_A\rangle + b|1_A\rangle. \tag{7.44}$$

Thus the total input state reads

$$\begin{aligned}
|\Psi\rangle_{DA} &= (\alpha\,|0_D\rangle \pm \beta\,|1_D\rangle) \otimes (a\,|0_A\rangle + b\,|1_A\rangle)\\
&= \alpha a\,|0_D0_A\rangle + \alpha b\,|0_D1_A\rangle \pm \beta a\,|1_D0_A\rangle \pm \beta b\,|1_D1_A\rangle.
\end{aligned} \tag{7.45}$$

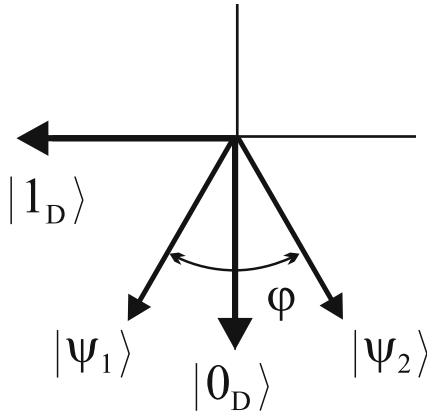

Fig. 13. The states $|\psi_1\rangle$ and $|\psi_2\rangle$ [defined by Eq. (7.44)] with real coefficients α and β can be visualized in a two-dimensional real space. The angle φ is related to the overlap of the two states: $\langle\phi_1|\phi_2\rangle = \cos\varphi = |\alpha|^2 + |\beta|^2 = 2|\alpha|^2 - 1$.

After the action of transformation (7.42) on this state one obtains the resulting state in the following form [the transformation is fixed for all φ; still $\cos\theta = \tan(\varphi_0/2)$]

$$\begin{aligned}\mathcal{U}_{DA}|\Psi\rangle_{DA} = &(\alpha a\cos\theta - \alpha b\sin\theta)\,|0_D 0_A\rangle \\ &+ (\alpha a\sin\theta + \alpha b\cos\theta)\,|0_D 1_A\rangle \\ &\pm\beta a\,|1_D 0_A\rangle \pm \beta b\,|1_D 1_A\rangle. \end{aligned} \tag{7.46}$$

If the coefficients a and b in the state of the ancilla are chosen in such a way that

$$\mathcal{U}_{DA}|\Psi\rangle_{DA} = (\alpha a\cos\theta - \alpha b\sin\theta) = \beta a := q/\sqrt{2} \tag{7.47}$$

then the expression (7.46) simplifies to the form

$$\mathcal{U}_{DA}|\Psi\rangle_{DA} = q\,|\pm\rangle + \text{const}_1\,|0_D 1_A\rangle \pm \text{const}_2\,|1_D 1_A\rangle, \tag{7.48}$$

where the states $|\pm\rangle$ are defined by Eq. (7.42). Clearly, applying the projective measurement introduced above one is able to discriminate unambiguously states (7.44) for any given $\varphi \in (0, \pi)$ provided he/she has prepared the proper state of the ancilla. The first term in Eq. (7.48) corresponds

to the successful discrimination, while the last two terms correspond to inconclusive results. The probability of success is

$$P_{\text{succ}} = |q|^2 = P_{\text{opt}} R(\varphi, \varphi_0) = 2 \sin^2 \frac{\varphi}{2} R(\varphi, \varphi_0), \tag{7.49}$$

where

$$R(\varphi, \varphi_0) = \frac{\cos \varphi_0 \, (\cos \varphi + 1)}{1 + \cos \varphi_0 - \sin \varphi \sin \varphi_0} \tag{7.50}$$

is the ratio between the actual value of the probability of successful discrimination and its optimal value. This expression is obtained from the condition (7.47) together with the normalization relation $|a|^2 + |b|^2 = 1$.

From above it follows that it is possible to implement a "universal quantum multi-meter" that is able to discriminate probabilistically but unambiguously (with no errors) between two non-orthogonal states for the large class of non-orthogonal pairs. The selection of the desired regime (i.e., the selection of the pair of states that should be unambiguously discriminate) is done by the choice of the quantum state of the ancillary qubit. This program state selects the measurement to be performed on the system. The probability of the successful discrimination can be optimal only for one such pair of states.

In the limit case when $\varphi_0 = 0$, i.e., $\theta = \pi/2$ (this is the fixed parameter of the employed unitary transformation), the probability of the successful discrimination for different φ's (i.e., for different settings of the ancilla and different pairs of input states) is the same as in the "quasi-classical" case, $P_{\text{succ}} = \frac{1}{2} \sin^2 \varphi$. By a quasi-classical approach we mean the probabilistic measurement when one randomly selects[11] the projective measurement in one of two orthogonal basis that both span the two-dimensional space containing both non-orthogonal states of interest (7.44). One basis consists of the state $|\psi_1\rangle$ and its orthogonal complement $|\psi_1^{\perp}\rangle$. If one finds the result corresponding to $|\psi_1^{\perp}\rangle$ he/she can be sure that the state $|\psi_1\rangle$ was not present. Analogously, the other basis consists of the state $|\psi_2\rangle$ and its orthogonal complement.

On the other hand, when $\varphi_0 = \pi/2$, i.e., $\theta = 0$, there is no way how to fulfill the condition (7.47) with $a \neq 0$ (and $P_{\text{succ}} \neq 0$) unless $\alpha = \beta = 1/\sqrt{2}$. That is, only two orthogonal states (7.42) can be unambiguously discriminated.

If the parameter φ_0 is somewhere in between 0 and $\pi/2$ the probability of success (as a function of φ) is very close to the optimal value in

[11] With the same probabilities provided that the frequencies of the occurrence of the input states are also the same.

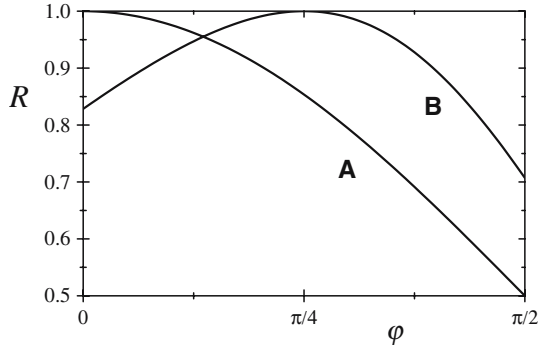

Fig. 14. The ratio $R(\varphi, \varphi_0)$ of the actual probability of successful discrimination to the optimal value of this probability as a function of the angle φ between two considered state vectors. The curve (A) shows the "quasi-classical" limit ($\varphi_0 = 0$). The curve (B) represents the case when $\varphi_0 = \pi/4$.

the relatively large vicinity of φ_0; see Fig. 14. However, for small values of φ it goes below the success probability of the quasi-classical case and for $\varphi = \pi/2$ (orthogonal states) the probability of successful discrimination is lower than unity.

One can ask for the optimal value of φ_0 in the sense that the average probability of successful discrimination [or, alternatively, function $R(\varphi, \varphi_0)$] over some chosen interval of φ's is maximal. For example, if we are interested in the average value of $R(\varphi, \varphi_0)$ over the interval of φ from 0 to $\pi/2$ we find that it is maximized when $\varphi_0 \approx 0.235\,\pi$ (the corresponding average value of R is 0.92).

For pedagogical reasons till now we have only worked with the states from a particular real subspace of the Hilbert space of the data qubit. However, it should be stressed that the method works for any two "input" states that are symmetrically displaced with respect to $|0_D\rangle$. In other words, the condition (7.47) can be fulfilled for any complex α and β. Simply,

$$\frac{b}{a} = \frac{1}{\sin\theta}\left(\cos\theta - \frac{\beta}{\alpha}\right).$$

The probability of the successful discrimination of states then reads

$$P_{\mathrm{succ}} = \frac{2\sin\theta\,|\alpha\beta|^2}{1 - 2\cos\theta\,\Re(\alpha\beta)}, \tag{7.51}$$

where $\Re(\alpha\beta)$ denotes the real part of $\alpha\beta$.

Comments

Above we have proposed a programmable quantum measurement device for the error-free discrimination of two non-orthogonal states of qubit that works with a large set of pairs of states. The device can be set to discriminate unambiguously any two states that are symmetrically located around some fixed state [in the sense of Eq. (7.44)]. The setting is done through the state of a program register that is represented by another qubit. This means that the particular pair of states that can be unambiguously discriminated is specified by the state of a "program" qubit. Two possible input states of the "data qubit" that are in correspondence with the program setting are never wrongly identified but from time to time we can get an inconclusive result. The probability of successful discrimination is optimal only for one program setting. However, the device can be designed in such a way that the probability of successful discrimination is very close to the optimal value for a relatively large set of program settings. Let us stress the *quantum nature* of the "programming": The states of the program register that represent different programs can be *non-orthogonal*.

8. APPROXIMATE PROGRAMMABLE QUANTUM PROCESSORS

In this section we shall discuss processors that approximate sets of unitary operators. We shall show, for a given processor, how to select an optimal program vector to approximate a particular unitary operator. In addition, we shall give a lower bound on the number of dimensions the program space must have to approximate a set of unitary operators to a given level of accuracy. In the last part of the paper we will address the question of optimal programmability, i.e., which processor is the best in approximating all channels.

8.1. Optimal Program States

Let us consider a processor that acts on the Hilbert space $\mathcal{H}=\mathcal{H}_d\otimes\mathcal{H}_p$, where $\mathcal{H}_d$ is the data Hilbert space and $\mathcal{H}_p$ is the program Hilbert space. Let us denote the dimension of $\mathcal{H}_d$ by D and that of $\mathcal{H}_p$ by N. The processor itself is represented by a unitary operator G, which acts on $\mathcal{H}$. The action of the processor on the input state $|\psi\rangle_d|\Xi\rangle_p$ is given by[9]

$$G(|\psi\rangle_d\otimes|\Xi\rangle)=\sum_{j=1}^{N}A_j(\Xi)|\psi\rangle_d|j\rangle_p, \tag{8.1}$$

where $\{|j\rangle_p | j = 1, \ldots, N\}$ is an orthonormal basis of $\mathcal{H}_p$. The operators $A_j(\Xi)$ are expressed in terms of the operators A_{jk}, where G is expressed as

$$G = \sum_{j,k=1}^{N} A_{jk} \otimes |j\rangle_p \, {}_p\langle k|. \tag{8.2}$$

These operators obey the relations

$$\sum_{j,k=1}^{N} A_{jk_1}^{\dagger} A_{jk_2} = I_d \delta_{k_1 k_2};$$
$$\sum_{k=1}^{N} A_{j_1 k}^{\dagger} A_{j_2 k} = I_d \delta_{j_1 j_2}, \tag{8.3}$$

where I_d is the identity operator on $\mathcal{H}_d$. The operator $A_j(\Xi)$ is given by

$$A_j(\Xi) = \sum_{k=1}^{N} A_{jk} \, {}_p\langle k|\Xi\rangle_p, \tag{8.4}$$

from which it follows that

$$\sum_{j=1}^{N} A_j^{\dagger}(\Xi) A_j(\Xi) = I_d. \tag{8.5}$$

We now need to discuss how to measure how close our processor comes to achieving a particular unitary operation. We shall use, what has been called by Gilchrist *et al*., the process fidelity,[68] which was originally proposed by Raginsky.[69,70] It is defined as follows. Let T_1 and T_2 be two completely positive maps, which map operators on the Hilbert space $\mathcal{K}$ onto operators on the same space. We shall assume that the dimension of $\mathcal{K}$ is finite and equal to D. The Jamiolkowski isomorphism allows us to associate a density matrix on $\mathcal{K} \otimes \mathcal{K}$ with each of these maps. Define the maximally entangled state

$$|\Phi\rangle = \frac{1}{\sqrt{D}} \sum_{j=1}^{D} |j\rangle |j\rangle, \tag{8.6}$$

where $\{|j\rangle | j = 1, \ldots, N\}$ is an orthonormal basis of $\mathcal{K}$. For each map T_j, define the density matrix ρ_j to be

$$\rho_j = (\mathcal{I} \otimes T_j)(|\Phi\rangle\langle\Phi|), \tag{8.7}$$

for $j=1,2$, where $\mathcal{I}$ is the identity map. The process fidelity is defined as

$$F_{\text{proc}}(T_1,T_2)=\left[\text{Tr}\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right]^2. \tag{8.8}$$

The process fidelity has a number of useful properties that are discussed in Refs. 69, 70 and 68, one of which is the fact that it is symmetric, i.e., $F_{\text{proc}}(T_1,T_2)=F_{\text{proc}}(T_2,T_1)$.

We are going to be interested in the case in which one of the maps is unitary. In particular, let us assume that $T_1(\rho)=U\rho U^{-1}$ for some unitary operator U. In this case we have that ρ_1 is a pure state so that $\rho_1^{1/2}=\rho_1$. This gives us that

$$\text{Tr}\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}=\frac{1}{D}\left[\sum_{j_1,j_2=1}^{D}\langle j_1|U^{-1}T_2(|j_1\rangle\langle j_2|)U|j_2\rangle\right]^{1/2}. \tag{8.9}$$

If T_2 is the result of the action of a processor, we have for a density matrix ρ_d, representing a data state, that

$$T_2(\rho)=\sum_{j=1}^{N}A_j(\Xi)\rho_d A_j(\Xi)^\dagger, \tag{8.10}$$

which gives us, finally, that (we denote the map T_1 by the operator U)

$$F(U,T_2)=\frac{1}{D^2}\sum_{j=1}^{N}\left|\text{Tr}(U^{-1}A_j(\Xi))\right|^2. \tag{8.11}$$

Using the notation for the Hilbert–Schmidt scalar product $(A|B)=\text{Tr}A^\dagger B$ this can be rewritten in the form $F(U,T_2)=\frac{1}{D^2}\sum_j|(A_j(\Xi)|U)|^2$.

This fidelity can also be expressed in terms of the operators A_{jk}. Defining the matrix

$$M_{k_1k_2}=\frac{1}{D^2}\sum_{j=1}^{N}\text{Tr}\left(A_{jk_1}^\dagger U\right)\text{Tr}\left(U^{-1}A_{jk_2}\right), \tag{8.12}$$

we have, from Eq. (8.4), that

$$F(U,T_2)=\sum_{k_1,k_2=1}^{N}{}_p\langle\Xi|k_1\rangle_p M_{k_1k_2}\,{}_p\langle k_2|\Xi\rangle_p. \tag{8.13}$$

Now consider the following problem. Suppose we are given a processor and we wish to find the best program to approximate the unitary operator U, where by best we mean the program that maximizes the process fidelity. An examination of Eq. (8.13) shows that this can be accomplished by finding the eigenvector of $M = \sum_{k_1,k_2} M_{k_1k_2}|k_1\rangle\langle k_2|$ with the largest eigenvalue, and choosing the program vector to be this eigenvector. The corresponding fidelity will just be the largest eigenvalue of M.

This procedure is particularly simple to carry out when the processor is, what was called in Ref. 9, a U processor. This is a processor that is a controlled-U gate. Each basis vector $|k\rangle_p$ in $\mathcal{H}_p$ is associated with a unitary operator U_k acting on $\mathcal{H}_d$. That is, if the program state is $|k\rangle_p$, then the operator U_k is applied to the data state. The operators A_{jk} for this type of processor are particularly simple, $A_{jk} = \delta_{jk}U_k$, which implies that the matrix M is given by

$$M_{k_1k_2} = \frac{1}{D^2}\left|\mathrm{Tr}(\mathrm{U}^\dagger \mathrm{U}_{k_1})\right|^2 \delta_{k_1k_2}. \tag{8.14}$$

Because in this case M is diagonal, we simply find the diagonal element that is largest. This is the largest eigenvalue of M and the maximum value of the fidelity. The value of k corresponding to this diagonal element tells us which of the basis vectors $|k\rangle_p$ is the program that will achieve this fidelity. This implies that to best approximate a unitary operator U by a U processor, we simply find which of the unitary operators that the processor can perform perfectly has the largest Hilbert–Schmidt inner product with U and perform that operation. Note that this prescription does not make use of superpositions of the basis states in the processor.

8.2. An Example

Before proceeding with the exploration of the general properties of approximate quantum processors, it is useful to analyze the following example. We shall consider a processor acting on qubits with an N dimensional program space spanned by the orthonormal basis $\{|k\rangle_p | k = 0, \ldots, N-1\}$. Define the shift operators E_+ and E_-, acting on the program space as $E_+|k\rangle = |k+1\rangle$ and $E_-|k\rangle = |k-1\rangle$, where the addition and subtraction are modulo N. We also define the program states

$$|\theta\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{-ik\theta}|k\rangle. \tag{8.15}$$

If $\theta = \theta_m = (2\pi m)/N$ then the state $|\theta_m\rangle$ becomes an eigenstate of E_+ and E_-

$$E_+|\theta_m\rangle = e^{i\theta_m}|\theta_m\rangle; \quad E_-|\theta_m\rangle = e^{-i\theta_m}|\theta_m\rangle. \tag{8.16}$$

For the qubit, whose Hilbert space is spanned by the two orthonormal vectors $|0\rangle_d$ and $|1\rangle_d$, define the operators $\sigma^{(+)}$ and $\sigma^{(-)}$, where $\sigma^{(+)}|0\rangle_d = |1\rangle_d$, $\sigma^{(+)}|1\rangle_d = 0$, and $\sigma^{(-)} = (\sigma^{(+)})^\dagger$. We shall consider a specific realization of the U processor defined by the operator G acting on $\mathcal{H}_d \otimes \mathcal{H}_p$

$$G = \exp\left[i\left(\frac{\pi}{2}\right)\left(\sigma^{(+)} \otimes E_- + \sigma^{(-)} \otimes E_+\right)\right]. \tag{8.17}$$

The fact that G is a U processor can be seen when we let G to act on the state $|\psi\rangle_d|\theta_m\rangle_p$. Here we obtain the result

$$\begin{aligned} |\Omega_m\rangle &= G(|\psi\rangle_d \otimes |\theta_m\rangle_p) \\ &= \exp\left[i\left(\frac{\pi}{2}\right)\left(e^{-i\theta_m}\sigma^{(+)} + e^{i\theta_m}\sigma^{(-)}\right)\right]|\psi\rangle_d \otimes |\theta_m\rangle_p. \end{aligned} \tag{8.18}$$

Defining

$$U(\theta) = \exp\left[i\left(\frac{\pi}{2}\right)\left(e^{-i\theta}\sigma^{(+)} + e^{i\theta}\sigma^{(-)}\right)\right], \tag{8.19}$$

we see that we can perform $U(\theta)$ perfectly when $\theta = \theta_m$, for some m. Suppose, however, we are interested in using this processor to approximately perform $U(\theta)$, for θ not equal to any of the θ_m. We know what the optimal strategy is from the previous section, find the operator $U(\theta_m)$ which has the greatest overlap (in the sense of the Hilbert–Schmidt inner product) with $U(\theta)$ and perform that operation. Here we are going to examine a strategy, which is simpler to implement, but not optimal. We shall simply use the state $|\theta\rangle_p$ as a program state. We find that this gives us a process fidelity of

$$F = \frac{1}{N^2}\sum_{m=0}^{N-1}\cos^2(\theta_m - \theta)\frac{\sin^2[N(\theta_m - \theta)/2]}{\sin^2[(\theta_m - \theta)/2]}. \tag{8.20}$$

This sum is an oscillatory function of θ with a period $2\pi/N$. The minima of this function are achieved for $\theta = \pi/N + 2\pi k/N$ when the process fidelity takes the minimal value $F_{\min} = 1 - 2/N$.

Let us see how this compares to using the optimal program states. The process fidelity between the operators $U(\theta_1)$ and $U(\theta_2)$ is given by

$$F(U(\theta_1), U(\theta_2)) = \cos^2(\theta_1 - \theta_2). \tag{8.21}$$

If we approximate $U(\theta)$ by $U(\theta_m)$, where m is chosen so that $U(\theta)$ and $U(\theta_m)$ have the largest Hilbert–Schmidt inner product, then the fidelity is bounded below by

$$F \geq \cos^2\left(\frac{\pi}{N}\right) \sim 1 - \left(\frac{\pi}{N}\right)^2. \tag{8.22}$$

Note that in this case the error is of order $1/N^2$, while in the previous case it was of order $1/N$, so there is a cost to not using the best program states.

What we then have is a an approximate processor that can be made very accurate by choosing N large enough. It achieves an accuracy of order $1/N$ in approximating $U(\theta)$ with the simple program state $|\theta\rangle_d$, which is not as good as the best accuracy, $1/N^2$, but the approximation in none the less a good one for N sufficiently large. Thus, we see that a U processor, making use of a simple program, can be quite useful in approximating the action of a set of operators labeled by a continuous parameter.

8.3. Bound on Program Space Dimension

We would now like to find a bound on the resources required to achieve a given accuracy in approximating a set of unitary operators by means of a fixed processor. In particular, we want to see how the dimension of the program space grows as the accuracy of the approximation increases.

The Schwartz inequality $|(A|B)| \leq \sqrt{(A|A)(B|B)}$ implies that

$$|\mathrm{Tr}(U^\dagger A_j(\Xi))| \leq \sqrt{D}[\mathrm{Tr}(A_j^\dagger(\Xi)A_j(\Xi))]^{1/2}, \tag{8.23}$$

and, therefore, if the action of our processor with the program state $|\Xi\rangle_p$ is given by the map T, we have that

$$\begin{aligned} F(U,T) &= \frac{1}{D^2}\sum_{j=1}^{N} |\mathrm{Tr}(U^\dagger A_j(\Xi))|^2 \\ &\leq \frac{1}{D}\sum_{j=1}^{N} \mathrm{Tr}(A_j^\dagger(\Xi)A_j(\Xi)) = 1. \end{aligned} \tag{8.24}$$

In the last equality we used the normalization property of Kraus operators (8.5), i.e., $\sum_j A_j^\dagger(\Xi)A_j(\Xi) = I$.

We begin by assuming that the fidelity is 1 and seeing what this implies about the operators $A_j(\Xi)$. If $F(U,T) = 1$, then, we see from above, that Schwartz inequality has to be saturated. This means that the

operators $A_j(\Xi)$ and U are colinear, i.e., $A_j(\Xi)=\beta_j U$, where β_j is a complex number. Furthermore, Eq. (8.5) implies $\sum_{j=1}^{N}|\beta_j|^2=1$. Now suppose that we have two different unitary operators that can be realized perfectly, U_1 by the program state $|\Xi_1\rangle_p$ and U_2 by the program state $|\Xi_2\rangle$. Therefore, $A_j(\Xi_1)=\beta_{1j}U_1$ and $A_j(\Xi_2)=\beta_{2j}U_2$. We then have that

$$\begin{aligned}\sum_{j=1}^{N}\beta_{1j}^{*}\beta_{2j}U_1^{-1}U_2 &= \sum_{j=1}^{N}A_j^{\dagger}(\Xi_1)A_j(\Xi_2)\\ &= I_{d\ p}\langle\Xi_1|\Xi_2\rangle_p,\end{aligned} \tag{8.25}$$

where we have used Eqs. (8.4) and (8.3). If $U_1\neq U_2$, then this equation implies that both ${}_p\langle\Xi_1|\Xi_2\rangle_p$ and $\sum_{j=1}^{N}\beta_{1j}^{*}\beta_{2j}$ are zero. This result is simply a restatement of the Nielsen–Chuang theorem: If two unitary operators are realized perfectly by a processor, their program vectors must be orthogonal.

Now let us suppose that the processor performs the operation U with a fidelity greater than or equal to $1-\epsilon$, i.e., $F(U,T)\geq 1-\epsilon$, where T is specified by Kraus operators $A_j(\Xi)$. Let us express these operators as

$$A_j(\Xi)=\beta_j U+B_j(\Xi), \tag{8.26}$$

where $\mathrm{Tr}(U^{\dagger}B_j(\Xi))=0$. This decomposition is unique. The inequality $F(U,T)\geq 1-\epsilon$ implies the following condition on coefficients $\beta_j=\frac{1}{D}(U|A_j(\Xi))$

$$1\geq F(U,T)=\frac{1}{D^2}\sum_{j=1}^{N}|(U|A_j(\Xi))|^2=\sum_{j=1}^{N}|\beta_j|^2\geq 1-\epsilon. \tag{8.27}$$

Tracing both sides of the normalization condition $\sum_j A_j(\Xi)^{\dagger}A_j(\Xi)=I$ we obtain the inequality $\sum_j \mathrm{Tr}[B_j(\Xi)^{\dagger}B_j(\Xi)]=\sum_j(B_j(\Xi)|B_j(\Xi))\leq D\epsilon$.

Next consider the situation in which our processor can approximate two unitary operators, U_1 and U_2, each with a fidelity greater than or equal to $1-\epsilon$. In particular, if T_1 is the map produced by the program state $|\Xi_1\rangle_p$ and T_2 is the map produced by the program state $|\Xi_2\rangle_p$, then both $F(U_1,T_1)$ and $F(U_2,T_2)$ are greater than or equal to $1-\epsilon$. We also have that

$$\begin{aligned}A_j(\Xi_1) &= \beta_{1j}U_1+B_{1j}(\Xi_1);\\ A_j(\Xi_2) &= \beta_{2j}U_2+B_{2j}(\Xi_2),\end{aligned} \tag{8.28}$$

where $\mathrm{Tr}(U_1^\dagger B_{1j}(\Xi_1)) = \mathrm{Tr}(U_2^\dagger B_{2j}(\Xi_2)) = 0$. As in the case when the unitary operators were performed perfectly, consider the quantity

$$\begin{aligned} I_d\langle\Xi_1|\Xi_2\rangle &= \sum_{j=1}^{N} A_j(\Xi_1)^\dagger A_j(\Xi_2) \\ &= \sum_{j=1}^{N} [\beta_{1j}^* U_1^\dagger + B_{1j}^\dagger(\Xi_1)][\beta_{2j} U_2 + B_{2j}(\Xi_2)]. \end{aligned} \tag{8.29}$$

Let us evaluate the absolute value of traces of both sides

$$\begin{aligned} D|\langle\Xi_1|\Xi_2\rangle| &= \left|\sum_j (A_j(\Xi_1)|A_j(\Xi_2))\right| \\ &= \Bigg|\sum_j [\beta_{1j}^*\beta_{2j}(U_1|U_2) + \beta_{1j}^*(U_1|B_{2j}) \\ &\qquad + \beta_{2j}(B_{1j}|U_2) + (B_{1j}|B_{2j})]\Bigg| \\ &\le |(U_1|U_2)|\sum_j \beta_{1j}^*\beta_{2j}| + 2D\sqrt{\epsilon} + D\epsilon. \end{aligned} \tag{8.30}$$

In the last line we used the formulas

$$\begin{aligned} \sum_j |(B_{1j}|B_{2j})| &\le \sum_j \sqrt{(B_{1j}|B_{1j})(B_{2j}|B_{2j})} \\ &\le \sqrt{\sum_j (B_{1j}|B_{1j}) \sum_j (B_{2j}|B_{2j})} \\ &\le D\epsilon, \end{aligned} \tag{8.31}$$

and

$$\begin{aligned} \left|\sum_j \beta_{1j}^*(U_1|B_{2j})\right| &\le \sum_j |\beta_{1j}|\sqrt{(U_1|U_1)(B_{2j}|B_{2j})} \\ &\le D\sqrt{\epsilon}\,. \end{aligned} \tag{8.32}$$

As a result we obtain the bound on the inner product between two program states

$$|\langle\Xi_1|\Xi_2\rangle| \le \frac{1}{D}|(U_1|U_2)|\left|\sum_j \beta_{1j}^*\beta_{2j}\right| + 2\sqrt{\epsilon} + \epsilon\,. \tag{8.33}$$

Next we will estimate the first term. The idea is to use Eq.(8.29) and apply both sides to a special vector $|\psi_\eta\rangle$ that maximizes the quantity $1 - |\langle\psi|U_1^\dagger U_2|\psi\rangle|^2$. Let us denote this maximum by η, i.e.,

$$\eta = \max_\psi \left[1 - |\langle\psi|U_1^\dagger U_2|\psi\rangle|^2\right]. \tag{8.34}$$

This quantity describes the distinguishability of two unitary transformations, and a short calculation shows that $\eta \le \|U_1 - U_2\|^2$. After applying both sides of Eq. (8.29) to $|\psi_\eta\rangle$ we find the components of the resulting vectors orthogonal to $|\psi_\eta\rangle$ by applying the projection operator $P_\eta^\perp = I - |\psi_\eta\rangle\langle\psi_\eta|$ to both sides. The left side vanishes and we obtain the equality

$$\begin{aligned} 0 &= P_\eta^\perp \Big(\sum_j A_j(\Xi_1)^\dagger A_j(\Xi_2)\Big)|\psi_\eta\rangle \\ &= \sum_j \beta_{1j}^* \beta_{2j} P_\eta^\perp U_1^\dagger U_2|\psi_\eta\rangle + |\omega\rangle, \end{aligned} \tag{8.35}$$

where

$$\begin{aligned} |\omega\rangle = P_\eta^\perp \sum_{j=1}^N \Big(&\beta_{1j}^* U_1^\dagger B_{2j}(\Xi_2) + \beta_{2j} B_{1j}^\dagger(\Xi_1) U_2 \\ &+ B_{1j}^\dagger(\Xi_1) B_{2j}(\Xi_2)\Big)|\psi_\eta\rangle. \end{aligned} \tag{8.36}$$

We now want to find a bound on $\|\omega\|$. Using the facts that the operator norm is bounded by the Hilbert–Schmidt norm, we have that

$$\begin{aligned} \left\|\sum_{j=1}^N B_j^\dagger(\Xi_1) B_j(\Xi_2)\right\| &\le \sum_{j=1}^N (B_j(\Xi_1)|B_j(\Xi_1))^{1/2} \\ &\quad \times (B_j(\Xi_2)|B_j(\Xi_2))^{1/2} \\ &\le \epsilon D, \end{aligned} \tag{8.37}$$

and

$$\begin{aligned} \left\|\sum_{j=1}^N \beta_{1j}^* U_1^\dagger B_{2j}(\Xi_2)\right\| &\le \sum_{j=1}^N |\beta_{1j}| (B_j(\Xi_2)|B_j(\Xi_2))^{1/2} \\ &\le \sqrt{\epsilon D}. \end{aligned} \tag{8.38}$$

Applying these inequalities we have that $||\omega|| \leq \epsilon D + 2\sqrt{\epsilon} D$. In addition, we find that $||P_\eta^\perp U_1^\dagger U_2 \psi_\eta|| = \sqrt{\eta}$. Therefore, we can conclude

$$\left| \sum_{j=1}^{N} \beta_{1j}^* \beta_{2j} \right| \leq \frac{\epsilon D + 2\sqrt{\epsilon D}}{\sqrt{\eta}}. \tag{8.39}$$

Defining

$$F = \min\left(1, \frac{\epsilon D + 2\sqrt{\epsilon D}}{\eta}\right), \tag{8.40}$$

we have, finally, that

$$|\langle \Xi_1 | \Xi_2 \rangle| \leq \frac{F}{D} |(U_1|U_2)| + 2\sqrt{\epsilon} + \epsilon. \tag{8.41}$$

Note that in the case that both operations are carried out without error, in which case $\epsilon = 0$, this inequality implies that the program vectors must be orthogonal, recovering the known result.

Now suppose that we have M unitary operators that we want implemented by a processor so that the process fidelity for each of the operators is greater than or equal to $1 - \epsilon$. How many dimensions must $\mathcal{H}_p$ have? In order to answer this question, we first find the values of $Y_{jk} = (F/D)|(U_j|U_k)|$ corresponding to each pair of operators in our set, and use these values to find the largest set of linearly independent vectors in the set of program vectors. Linear independence can be deduced from the following result: If $\{v_k | k = 1, \ldots, K\}$ are vectors of length 1, and $|\langle v_{k_1} | v_{k_2} \rangle| < 1/(K-1)$, then the vectors $\{v_k | k = 1, \ldots, K\}$ are linearly independent.[9,11] Suppose that there is a subset of our operators, with M' members, whose pairs have small values of Y_{jk}, and let the largest value of Y_{jk} for this subset be $Y_{\max}$. Then we have for all of the program vectors corresponding to this set, that

$$|\langle \Xi_j | \Xi_k \rangle| \leq Y_{\max} + 2\sqrt{\epsilon} + \epsilon = q(Y_{\max}, \epsilon). \tag{8.42}$$

Let K_q be the largest integer such that $K_q < (1/q) + 1$. What the result we just quoted implies, is that any set of vectors whose size is K_q or less, will be linearly independent. Therefore, if $M' \leq K_q$, then all of the program vectors will be linearly independent, and the dimension of $\mathcal{H}_p$ must be at least M'. If $M' > K_q$, then the dimension of $\mathcal{H}_p$ must be at least K_q. This, then, is the restriction our result imposes on the dimension of the program space.

As an example, suppose we want to implement the operators I, σ_1, σ_2, and σ_3 on qubits, where the operators σ_j, for $j=1,2,3$, correspond to the usual Pauli matrices. For all pairs of these operators we find that $Y_{jk}=0$, and

$$q(0,\epsilon)=2\sqrt{2\epsilon}+\epsilon. \tag{8.43}$$

Our bounds then give us that for $\epsilon<0.02$ the program space must have four dimensions, for $\epsilon<0.05$ it must have at least three dimensions, and for $\epsilon<0.17$ it must have at least two dimensions.

8.4. One-Parameter Group: Two Approaches

Programmable processors can be exploited to implement quantum maps probabilistically. In this case a specific measurement on the program state is performed and if an *a priori* defined result is obtained then we know that a desired operation has been performed on the data. In other words the specific measurement that is accompanied by a post-selection induces the desired transformation of the data register. As was discussed in Ref. 11 a probabilistic processor without measurement can be used as an approximate processor. In this case the transformation can be expressed as

$$\mathcal{E}_\xi[\varrho]=p_{\text{success}}\mathcal{T}[\varrho]+p_{\text{error}}\mathcal{N}[\varrho], \tag{8.44}$$

where $\mathcal{T}$ is the channel we want to approximate, and p_{success} and p_{error} are independent of the input data state, ρ. Due to the concavity of the square root of the process fidelity we find that $p_{\text{success}}\leq F(\mathcal{E}_\xi,\mathcal{T})$, i.e., the accuracy of the approximation is bounded from below by the probability of success.

Here we want to compare the performance of a probabilistic processor used as an approximate one with a different type of approximate processor in order to see which requires greater resources. Both will be used to implement operators in the same one-parameter group. In particular, consider the operations on qudits (with orthonormal basis $\{|k\rangle|k=1,\ldots,D\}$) specified by

$$U(\theta)=e^{i\theta}|1\rangle\langle 1|+X, \tag{8.45}$$

where $X=\sum_{k=2}^{N}|k\rangle\langle k|$, and $0\leq\theta<2\pi$.

Consider the processor described by the operators A_{jk} for $1 \leq j, k \leq N$, where

$$A_{jk} = \begin{cases} \delta_{jk} X + \delta_{k,j+1} |1\rangle\langle 1| & j < N; \\ \delta_{Nk} X + \delta_{k,1} |1\rangle\langle 1| & j = N, \end{cases} \tag{8.46}$$

originally described in Ref. 42. With the program state

$$|\Xi\rangle = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} e^{i(k-1)\theta} |k\rangle, \tag{8.47}$$

we find that for $1 \leq j \leq N-1$

$$A_j(\Xi) = \frac{1}{\sqrt{N}} e^{i(j-1)\theta} U(\theta), \tag{8.48}$$

and for $j = N$

$$A_N(\Xi) = \frac{1}{\sqrt{N}} \left(e^{i(N-1)\theta} X + |1\rangle\langle 1| \right). \tag{8.49}$$

What this means is that if after the action of the processor, the program state is measured in the basis $\{|1\rangle_p, \ldots |N\rangle_p\}$ and if the result $|j\rangle_p$ is obtained, where $j \neq N$, then the operation $U(\theta)$ has been carried out on the data. However, if the result $|N\rangle_p$ is obtained, then the operation $U(\theta)$ has not been performed. Because each of these outcomes is equally likely, the probability of obtaining the desired result is $(N-1)/N$. If instead of measuring the output of the program register we discard it, i.e., trace over it, we can use this processor as an approximate one. The process fidelity in this case is given by

$$F = 1 - \frac{2(D-1)}{ND^2}(1 - \cos(N\theta)). \tag{8.50}$$

Another processor that will approximate this one-parameter group can be constructed by dividing the interval $[0, 2\pi)$ into subintervals and approximating all of the operators $U(\theta)$ for θ in a particular subinterval by a single operator. In particular, let $\Delta\theta = \pi/N$, and approximate $U(\theta)$ for $2j\Delta\theta \leq \theta \leq 2(j+1)\Delta\theta$ by $U_j = U((2j+1)\Delta\theta)$, where $j = 0, 1, \ldots, N-1$. We now define a U processor by setting, for $j, k = 0, 1, \ldots, N-1$

$$A_{jk} = \delta_{jk} U_j. \tag{8.51}$$

In order to approximate $U(\theta)$ for $2j\Delta\theta \le \theta \le 2(j+1)\Delta\theta$, we choose the program state $|\Xi\rangle_p = |j\rangle_p$. For this processor we find that

$$1 - F \le \frac{2(D-1)}{D^2}(1-\cos\Delta\theta) \sim \frac{2(D-1)}{D^2}\frac{\pi^2}{4N^2}. \tag{8.52}$$

By comparing the two fidelities, we see that for a fixed value of the program space dimension, N, the second processor will provide a greater accuracy.

Comments

In this section we have examined the approximation of a set of unitary operators by means of a programmable quantum circuit, i.e., a quantum processor. The programs themselves are quantum states. We have shown, for a fixed processor, how to find the program that induces the best approximation of a particular unitary operator. In addition, we have found bounds on the size of the program space that is necessary to approximate a set of operators to a given precision.

Approximate processors can be characterized by their accuracy and by the resources they require. By the accuracy, or level of precision, we mean the quantity $\epsilon_G = 1 - \min_{\mathcal{E}\in\Gamma}\max_{\xi\in\mathcal{S}(\mathcal{H}_p)} F(\mathcal{E}, \mathcal{E}_\xi)$.[12] Here Γ is the set of maps we want to realize, $\mathcal{S}(\mathcal{H}_p)$ is the set of positive operators on $\mathcal{H}_p$ with trace one (note that we are allowing mixed program states here) and $\mathcal{E}_\xi[\varrho] = \mathrm{Tr}_p G\varrho \otimes \xi G^\dagger$. The dimension of the program space, N, characterizes the resources required. We wish to know how these two parameters are related. We have made some progress here in exploring this relation for limited sets of maps. The problem becomes more difficult if one considers Γ to be the set of all unitary maps and harder yet if it is the set of all completely positive trace-preserving maps. Once we have these definitions of precision and resources, we can consider two problems. First, given a specific degree of precision ε_G for some set of maps Γ, how large must the program space be? Second, for fixed resources, what is the optimal processor, i.e., for which G is the accuracy the best (ε_G the least)? In Ref. 71 one case of this problem was solved by D'Ariano and Perinotti. The data states were qubits, and Γ was the set of unitary operators acting on a single qubit. The program space was also a single quibit so that $N = 2$. They then showed that the optimal accuracy is given by $\varepsilon_G = 3/4$. This precision can be achieved when G is a swap gate,(72) i.e., $C_{\mathtt{SWAP}}(|\psi\rangle \otimes |\phi\rangle) = |\phi\rangle \otimes |\psi\rangle$ for all states $|\psi\rangle, |\phi\rangle$. For a processor with both the data and program spaces having the same dimension D and G

[12] The definition of the accuracy of the processor ε_G is not unique. For instance, one can consider the CB-norm and use averages in the definition.

given by the d-dimensional version of the swap gate, we find that $F(U, \mathcal{E}_\xi)$, where $\mathcal{E}_\xi$ is the map induced on the data by the processor with program ξ, is independent of both the program and U, and is equal to $1/D^2$. This implies that for this processor, the accuracy is given by $\varepsilon_G = 1 - \frac{1}{D^2}$. We suspect that this is the optimal value if the size of the program register equals to the size of the data register, i.e., $N = D$, but whether this suspicion is correct is beyond the scope of this paper and will be analyzed elsewhere.[72]

There are many open issues remaining. One possibility is to shift our focus, and rather than ask what type of the processor can perform a given set of operations with a particular level of precision, ask instead if it is possible to characterize the operations that a given processor can perform to a specified accuracy. Another issue is the following. So far, we have assumed that we are approximating a set of unitary operators with just a single use of a processor. What happens if we can use the same processor more than once? It turns out that multiple usage of the processor can significantly improve the accuracy of the approximation. In particular, when the U processor (which can perform a set of unitary operators perfectly) is used n times, the one can perfectly perform not only the original set of operators, but any product of these operators that is of length n or less.

It would also be useful to find specific processors, which are not U processors, that can approximate a wide class of unitary operations. As we have seen, superpositions of the basis program states are not useful in optimally approximating a unitary operator with a U processor, but they very well may be useful in doing so with other types of processors.

Probabilistic processors have shown themselves to be very flexible devices. They can perform large classes of operations while requiring only limited resources. Their drawback is that these operations are performed with a probability that is less than one. It remains to be seen how flexible deterministic processors are, but the results here place some constraints on what they can accomplish. In this paper we have given an example of how a probabilistic processor can be used as an approximate one.

9. CONCLUSION

We have introduced a concept of programmable quantum devices. These devices take as an input two registers: the data and the program register. A quantum state of the program register contains information about the transformation to be performed on the data register. The usefulness of this set up is obvious when we consider a situation when a set of instructions that characterize an operation to be performed on the data

is encoded in a *single* copy of a quantum system. This may happen when the set of instructions (a program) is obtained as an output of a quantum computer (whatever this device is). This output state might be in general *unknown*. In this situation one has two options: Firstly, one can measure and estimate the program state and with the obtained classical information one can perform a *classical* control of the evolution of the data register. The main obstacle in this approach is that the fidelity of estimation of a state of quantum system based on a measurement of just a single copy of the state is negligible small (it is inversely proportional to a dimension of the Hilbert space of the program register. This is the reason why the programmable quantum processor that takes as an input the unknown quantum program register is a better alternative. The quantum processor will perform operations that are specified by the program register even though a (classical) user of the processor does not have an information about the set of instructions.

In the paper we have analyzed various aspects of programmable quantum devices. We have considered deterministic as well as probabilistic programmable quantum processors. We have shown that probabilistic processors are universal, though the price to be paid for the universality is that the probability success is inversely proportional to the dimension of the data register. Luckily enough, there exists a solution to this problem—errors in implementation of probabilistic processors can be compensated via multiple run of the processor with error-correcting program states. We have shown how one can systematically improve performance of probabilistic quantum processors.

Through the paper we have been referring many times to the QID. This remarkably simple quantum logical network composed of four C-NOT gates has a potential of realizing many tasks in quantum information processing. Depending on the state of the program register the QID can be used for quantum cloning or for performing a universal NOT gate. It can serve as a universal probabilistic programmable processor, it can be used to perform specific POVM measurements. It is a challenge to realize this quantum "machine" experimentally.

ACKNOWLEDGMENTS

This research was supported in part by the European Union projects QGATES, QAP and CONQUEST, by the Slovak Academy of Sciences via the projects VEGA and CE-PI under the contract I/2/2005, and by the project APVT-99-012304. VB thanks the Alexander von Humboldt Foundation for support. We thank Adam Brazier, Peter Knight, and Matyas Koniorczyk for fruitful discussions and collaboration.

APPENDIX A: PREPROCESSING SUCCESS PROBABILITY

We have seen how the preprocessing scheme works for $|\Xi_\theta\rangle^{\otimes 3}$ and $|\Xi_\theta\rangle^{\otimes 7}$, and that it produces the same probability for success as the one-shot and iterative schemes with the same starting states. The general scheme for preprocessing $2^X - 1$ copies of the basic program state, where X is an integer, is an extension of the method used in Sec. 5.1.2 (see Ref. 73). Given $|\Xi_\theta\rangle^{\otimes 2^X-1}$, the best VMC/HZB program state that can be produced is $|\Xi_\theta^{(X)}\rangle$, because the phases start at 0, rise in increments of $-i\theta$ and the largest phase in $|\Xi_\theta\rangle^{\otimes 2^X-1}$ is $-i(2^X-1)\theta$, which is also the biggest phase in $|\Xi_\theta^{(X)}\rangle$, where the phases also rise in increments of $-i\theta$ from a phase of 0. The strategy will be to permute the phases on the 2^{2^X-1} terms in $|\Xi_\theta\rangle^{\otimes 2^X-1}$, where the number of terms with each phase is binomially distributed, in a useful way and then measure the leftmost $M = 2^X - 1 - X$ qubits to project into a remainder X-qubit state which will be $|\Xi_\theta^{(X)}\rangle$ or some other state which, upon further measurements of leftmost remaining qubits, will be projected into $|\Xi_\theta^{(r)}\rangle$ where $r \in \{1, 2, \ldots X-1\}$, up to a global phase, as was the case in the examples in Sec. 5.1.2 for $X = 2$ and $X = 3$, i.e., the permutation achieves:

$$\frac{1}{\sqrt{2^{2^X-1}}} \sum_{j=0}^{2^{2X-1}} e^{i|j|\theta} |j\rangle \rightarrow \frac{1}{\sqrt{2^M}} \sum_{k=0}^{2^M-1} |k\rangle \otimes |k_\Xi\rangle \,. \tag{A1}$$

The X-qubit states $|k_\Xi\rangle$ are given by

$$|k_\Xi\rangle = \sum_{l=1}^{X} \sum_{t=0}^{2^{X-l}-1} a_{lt}^k \left(|t\rangle \otimes |\Xi_\theta^{(l)}\rangle \right), \tag{A2}$$

where the $|t\rangle$ are $(X-l)$-qubit computational basis states and normalization requires that

$$\sum_{l=1}^{X} \sum_{t=0}^{2^{X-l}-1} |a_{lt}^k|^2 = 1 \tag{A3}$$

and we note that not all of the a_l^k need be non-zero. In addition, these coefficients have to be such that the measurement outcomes subsequent to the initial M-qubit measurement are entangled with a particular eventual outcome, i.e., one of the $|\Xi^l\rangle$ so that if we measure the initial M qubits then carry out some more measurements, that the final measurement outcome $|t\rangle$ tells us what VMC/HZB program state we have.

The allocation of phases in the construction of the permutation is done in the same way as was shown in some detail for $|\Xi_\theta\rangle^{\otimes 7}$, which is to say, first one of each phase is allocated to the 2^X terms that will produce $|\Xi_\theta^{(X)}\rangle$ upon one outcome of the measurement of the M leftmost qubits. Following that, phases $-i\theta \ldots -2^{X-1}i\theta$ and $-i(2^{X-1}-1)\theta \ldots -i(2^X-2)\theta$ (that was $-i\theta$ to $-4i\theta$ and $-3i\theta$ to $-6i\theta$ in the $X=3$, $N=7$ example) are allocated to sets of 2^{X-1} terms until the phases $-i\theta$ and $-i(2^X-2)\theta$ are exhausted and then phases $-2i\theta \ldots -(2^{X-1}+1)i\theta$ and $-(2^{X-1}-2)i\theta \ldots -i(2^X-3)$ are allocated, etc, until there are only $2^{X-1}-2$ different phases left available (the "middle" $2^{X-1}-2$ phases if laid out as in the tables of Sec. 5.1.3). These groups of terms will be those that realize $|\Xi_\theta^{(X-1)}\rangle$ post-measurement. Following this, the procedure is to allocate groups of 2^{X-2} phases so as to create groups of terms that will realize $|\Xi_\theta^{(X-2)}\rangle$ post-measurements, and so on, until the last remaining phases, $-i(2^{X-1}-1)\theta$ and $-2^{X-1}i\theta$, are allocated to the terms that will produce $|\Xi_\theta^{(1)}\rangle$ post-measurements.

The key facts here are that all of the phases can be allocated in this way to a group of terms associated, post-measurements, with the realization of a state $|\Xi(\theta)_s\rangle_p$ where $s \leq X$, as a little thought will show. Furthermore, with the phases allocated in this way, every group of phases allocated contains the "middle" two phases, $-i(2^{X-1}-1)\theta$ and $-2^{X-1}i\theta$. Thus, the number of groups of phases, W, is equal to the number of terms in $|\Xi_\theta\rangle^{\otimes 2^X-1}$ that have phase $-i(2^{X-1}-1)\theta$ or $-2^{X-1}i\theta$, i.e.,

$$W = \binom{2^X-1}{(2^X-2)/2}. \tag{A4}$$

If the number of groups corresponding to $|\Xi(\theta)_s\rangle_p$ is W_s, then, because each individual phase from the terms in $|\Xi_\theta\rangle^{\otimes 2^X-1}$ is allocated to one of these groups,

$$\sum_{s=1}^{X} W_s = W = \binom{2^X-1}{(2^X-2)/2}. \tag{A5}$$

Additionally, because all of the 2^{2^X-1} terms in $|\Xi_\theta\rangle^{\otimes 2^X-1}$ end up permuted into one of these sets, and because each set of form $|\Xi_\theta^{(s)}\rangle$ contains 2^s terms, with W_s sets of form $|\Xi_\theta^{(s)}\rangle$ and s different types of set, then

$$\sum_{s=1}^{X} 2^s W_s = 2^{(2^X-1)}. \tag{A6}$$

The probability, q_s, that the final result is $|\Xi_\theta^{(s)}\rangle$ following measurement(s), can be expressed in terms of W_s. It is equal to the number of terms that belong in sets of form $|\Xi_\theta^{(s)}\rangle$ divided by the total number of terms, i.e.,:

$$q_s = \frac{2^s W_s}{2^{(2^X-1)}}. \tag{A7}$$

Each state $|\Xi_\theta^{(s)}\rangle$ will, if it is the outcome of the calculation, succeed in the VMC/HZB scheme with a probability p_s given by:

$$p_s = 1 - \frac{1}{2^s} \tag{A8}$$

from Eq. 5.7.

The total success probability, p_X, from preprocessing $|\Xi_\theta\rangle^{\otimes 2^X-1}$ followed by the input of the resulting state as the program state into the HZB/VMC scheme, is

$$\begin{aligned} p_X &= \sum_s^X p_s q_s \\ &= \frac{1}{2^{(2^X-1)}}\left(\sum_{s=1}^X 2^s W_s - \sum_{s=1}^X W_s\right) \\ &= 1 - \frac{1}{2^{(2^X-1)}}\binom{2^X-1}{(2^X-2)/2}. \end{aligned} \tag{A9}$$

where the last step was achieved using Eqs. (A5) and (A6). The total number of basic program qubits, N, is given by:

$$N = 2^X - 1 \tag{A10}$$

and substituting this into Eq. (A9), the overall probability of success, p, is given by:

$$p = 1 - \frac{1}{2^N}\binom{N}{(N-1)/2}. \tag{A11}$$

This is the same result as for the single-shot and iterative procedures on $|\Xi_\theta\rangle^{\otimes N}$ and so preprocessing gives the same overall probability of success as in those case and the result is proved. Although this calculation is based on a specific method of allocation of the states, it will be true for any permutation allocation that puts all of the phases in the state $|\Xi_\theta\rangle^{\otimes 2^X-1}$ into a grouping that produces a state $|\Xi(\theta)_s\rangle_p$, $s \leq X$ and in which each grouping contains the two "middle" phases, i.e., the phases $-i(2^{X-1}-1)\theta$ and $-2^{X-1}i\theta$.

REFERENCES

1. S. Lloyd and L. Viola, *Phys. Rev. A*, **65**, 010101 (2002).
2. See for example: M. A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
3. G. Harel and V. M. Akulin, *Phys. Rev. Lett*. **82**, 1 (1999).
4. E. Brion, V. M. Akulin, D. Comparat, I. Dumer, V. Gershkovich, G. Harel, G. Kurizki, I. Mazets, and P. Pillet, *Non-Holonomic Control I.* `arXiv: quant-ph/0507156` (2005).
5. S. Lloyd and S. M. Braunstein, *Phys. Rev. Lett.* **82**, 1784 (1999).
6. B. Hladký, G. Drobný, and V. Bužek, *Phys. Rev. A* **61**, 0221202 (2000).
7. M. A. Nielsen and I. L. Chuang, *Phys. Rev. Lett.* **79**, 321 (1997).
8. M. Hillery, V. Bužek, and M. Ziman, *Fortschr. Phys.* **49**, 987 (2001).
9. M. Hillery, M. Ziman, and V. Bužek, *Phys. Rev. A* **66**, 042302 (2002).
10. J. Preskill, *Proc. Roy. Soc. Lond. A* **454**, 385 (1998).
11. G. Vidal, L. Masanes, and J. I. Cirac, *Phys. Rev. Lett.* **88**, 047905 (2002).
12. M. Hillery, V. Bužek, and M. Ziman, *Phys. Rev. A* **65**, 022301 (2002).
13. C. W. Helstrom, *Quantum Detection and Estimation Theory* (Academic Press, New York, 1976).
14. A.S. Holevo, *Probabilistic and Statistical Aspects of Quantum Theory* (North-Holland, Amsterdam, 1982).
15. A.Peres, *Quantum Theory: Concepts and Methods* (Kluwer, Dordrecht, 1993).
16. P. Busch, P. Lahti, and P. Mittalstead, *Quantum Theory of Measurement* (Springer, Berlin, 1996).
17. S. Massar and S. Popescu, *Phys. Rev. Lett.* **74**, 1259 (1995).
18. R. Derka, V. Bužek and A. K. Ekert, *Phys. Rev. Lett.* **80**, 1571 (1998).
19. S. F. Huelga, J. A. Vaccaro, and A. Chefles, *Phys. Rev. A* **63**, 042303 (2001).
20. J. Preskill, *Quantum Theory of Information and Computation*, see `http://www.theory.caltech.edu/people/ preskill`.
21. A. Uhlmann, *Rep. Math. Phys.* **9**, 273 (1976); *ibid* **24**, 229 (1986).
22. V. Scarani, M. Ziman, P. Štelmachovič, N. Gisin, and V. Bužek, *Phys. Rev. Lett.* **88**, 097905 (2002).
23. M. Ziman, P. Štelmachovič, V. Bužek, M. Hillery, V. Scarani, and N. Gisin, *Phys. Rev. A* **65**, 042105 (2002)].
24. S. Braunstein, V. Bužek, and M. Hillery, *Phys. Rev. A* **63**, 052313 (2001).
25. V. Bužek and M. Hillery, *Phys. Rev. A* **54**, 1844 (1996).
26. V. Bužek, S. Braunstein, M. Hillery, and D. Bruß, *Phys. Rev. A* **56**, 3446 (1997).
27. G. Alber, A. Delgado, N. Gisin, and I. Jex, Los Alamos arXive quant-ph/0008022.
28. V. Bužek, A. D. Wilson-Gordon, P. L. Knight and W. K. Lai, *Phys. Rev. A* **45**, 8079 (1992).
29. D. T. Pegg and S. M. Barnett, Europhys. Lett. **6**, 483 (1988); *Phys. Rev. A* **39**, 1665 (1989); D. T. Pegg, J. A. Vaccaro, and S. M. Barnett, *J. Mod. Opt.* **37**, 1703 (1990).
30. W. K. Wootters, *Ann. Phys.* **176**, 1 (1987).
31. T. Opatrný, V. Bužek, J. Bajer, and G. Drobný, *Phys. Rev. A* **52**, 2419 (1995).
32. U. Leonhardt, *Phys. Rev. Lett.* **74**, 4101 (1995).
33. M. Koniorczyk, V. Bužek, and J. Janszky, *Phys. Rev. A* **64**, 034301 (2001).
34. J. P. Paz, *Phys. Rev. A* **65**, 062311 (2002).
35. D. Galetti and A. F. R. de Toledo Piza, *Physica* **149A**, 267 (1988).
36. V. Bužek, C. H. Keitel, and P. L. Knight, *Phys. Rev. A* **51**, 2575 (1995).
37. G. Alber, A. Delgado, N. Gisin, and I. Jex, *J. Phys. A* **34**, 8821 (2001).
38. V. Vedral, A. Barenco, and A. Ekert, *Phys. Rev. A* **54**, 147 (1996).
39. A. D. Pittenger, *An Introduction to Quantum Computing* (Birkhäuser, Boston, 2000).

40. D. I. Fivel, *Phys. Rev. Lett.* **74**, 835 (1995).
41. M. Ziman and V. Bužek, *Int. J. Quant. Inf.*, **1**, 527 (2003).
42. M. Hillery, M. Ziman, and V. Bužek, *Phys. Rev. A* **69**, 042311 (2004).
43. M. Koniorczyk, V. Bužek, and P. Adam, *Eur. J. Phys. D* **37**, 275 (2006).
44. D. E. Evans and J. T. Lewis, *Dilations of Irreversible Evolutions in Algebraic Quantum Theory*, Communications of Dublin Institute of Advanced Studies, Series A (Theoretical Physics), No. 24, (DIAS, Dublin, 1977).
45. M. Dušek and V. Bužek, *Phys. Rev. A* **66**, 022112 (2002).
46. J. Fiurášek *et al.*, *Phys. Rev. Lett.* **89**, 190401 (2002).
47. J. Fiurášek and M. Dušek, *Phys. Rev. A* **69**, 032302 (2004).
48. J. A. Bergou and M. Hillery, *Phys. Rev. Lett.* **71**, 042314 (2005).
49. J. P. Paz and A. Roncaglia, *Phys. Rev. A* **68**, 052316 (2003).
50. A. K. Ekert, C. M. Alves, D. K. L. Oi, M. Horodecki, P. Horodecki, and L. C. Kwek, *Phys. Rev. Lett.* **88**, 217901 (2002).
51. G. M. D'Ariano, P. Perinotti, and M. F. Sacchi, *Europhys. Lett.* **65**, 165 (2004).
52. G. M. D'Ariano, P. Perinotti, and M. F. Sacchi, in *Proceedings of the 8th Int. Conf. on Squeezed States and Uncertainty Relations*, H. Moya-Cessa *et al.* (eds.) (Rinton, Princeton, 2003), p. 86.
53. G.M. D'Ariano and P. Perinotti, *Phys. Rev. Lett.* **93**, 180503 (2004)
54. M. Roško, V. Bužek, P. R. Chouha, and M. Hillery, *Phys. Rev. A* **68**, 062302 (2003).
55. J. M. Renes, R. Blume-Kohout, A. J. Scott, and C. M. Caves, *J. Math. Phys.* **45**, 2171 (2004).
56. P. Stelmachovič, *private communication*.
57. K. Wódkiewicz, *Phys. Rev. Lett.* **52**, 1064 (1984); *Phys. Lett. A* **115**, 304 (1986); Phys. Lett. A **129**, 1 (1988).
58. A. Chefles, *Contemp. Phys.* **41**, 401 (2001).
59. S. M. Barnett, *Fortschr. Phys.* **49**, 909 (2001).
60. I. D. Ivanovic, *Phys. Lett. A* **123**, 257 (1987).
61. D. Dieks, *Phys. Lett. A* **126**, 303 (1988).
62. A. Peres, *Phys. Lett. A* **128**, 19 (1988).
63. G. Jaeger and A. Shimony, *Phys. Lett. A* **197**, 83 (1995).
64. A. Chefles and S. M. Barnett, *Phys. Lett. A* **250**, 223 (1998).
65. B. Huttner, A. Muller, J. D. Gautier, H. Zbinden, and N. Gisin, *Phys. Rev. A* **54**, 3783 (1996).
66. J. A. Bergou, M. Hillery, and Y. Sun, *Fortschr. Phys.* **49**, 915 (2001).
67. M. Dušek, M. Jahma, and N. Lütkenhaus, *Phys. Rev. A* **62**, 022306 (2000).
68. A. Gilchrist, N. K. Langford, and M. A. Nielsen, *Phys. Rev. A* **71**, 062310 (2005).
69. M. Raginsky, *Phys. Lett. A* **290**, 11 (2001).
70. M. Raginsky, *Phys. Rev. A* **65**, 032306 (2002).
71. G. M. D'Ariano and P. Perinotti, quant-ph/0510033.
72. M. Hillery, M. Ziman, and V. Bužek, *Phys. Rev. A* **73**, 022345 (2006).
73. A. Brazier, V. Bužek, and P.L. Knight, *Phys. Rev. A* **71**, 032306 (2005).