Improving the performance of probabilistic programmable quantum processors

Mark Hillery,[1] Mário Ziman,[2] and Vladimír Bužek[2,3]
[1]*Department of Physics, Hunter College of CUNY, 695 Park Avenue, New York, New York 10021, USA*
[2]*Research Center for Quantum Information, Slovak Academy of Sciences, Dúbravská cesta 9, 845 11 Bratislava, Slovakia*
[3]*Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic*

(Received 5 November 2003; published 16 April 2004)

We present a systematic analysis of how one can improve performance of probabilistic programmable quantum processors. We generalize a simple Vidal-Masanes-Cirac processor that realizes U(1) rotations on a qubit with the phase of the rotation encoded in a state of the program register. We show how the probability of success of the probabilistic processor can be enhanced by using the processor in loops. In addition we show that the same strategy can be utilized for a probabilistic implementation of nonunitary transformations on qubits. In addition, we show that an arbitrary SU(2) transformation of qubits can be encoded in program state of a universal programmable probabilistic quantum processor. The probability of success of this processor can be enhanced by a systematic correction of errors via conditional loops. Finally, we show that all our results can be generalized also for qudits. In particular, we show how to implement SU(N) rotations of qudits via programmable quantum processor and how the performance of the processor can be enhanced when it is used in loops.

DOI: 10.1103/PhysRevA.69.042311 PACS number(s): 03.67.Lx

I. INTRODUCTION

Classical computers are programmable, that is, the task they perform is determined by a set of instructions that is sent into the machine along with the data to be processed. This is a very desirable feature; we do not have to build a different circuit every time we want to perform a different procedure. It would be useful to be able to develop quantum processors with the same property.

The development of programmable quantum circuits is an area that has attracted attention only recently. The basic model for these circuits consists of two parts: a data register and a program register. There are two inputs, a data state, which is sent into the data register and on which an operation is to be performed, and a program state, which is sent into the program register, that specifies the operation. The first result was due to Nielsen and Chuang, who showed that a deterministic *universal* quantum processor does not exist [1]. The problem is that a new dimension must be added to the program space for each unitary operator that one wants to be able to perform on the data. A similar situation holds if one studies quantum circuits that implement completely positive, trace-preserving maps rather than just unitary operators [2,3]. 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 be deterministic, then universal processors become possible [1,4–6]. These processors are probabilistic: they sometimes fail, but we know when this happens.

A number of examples of programmable quantum circuits have been proposed. One is a quantum "multimeter" that performs unambiguous state discrimination on a set of two states, the set being specified by the program [7]. There are also devices that evaluate the expectation value of an arbitrary operator, the data representing the state in which the expectation value is to be evaluated and the program state specifying the operator [8,9].

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 1 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.

In the case of one-parameter unitary groups acting on qubits this was done by Preskill [4] and Vidal, Masanes, and Cirac [5]. Vidal, Masanes, and Cirac considered the one-parameter group of operations given by $U(\alpha)=\exp(i\alpha\sigma_z)$, for $0\leqslant\alpha<2\pi$, and discussed two equivalent methods of making the probability of performing $U(\alpha)$ arbitrarily close to 1. A circuit consisting of a single controlled-NOT (CNOT) gate, with the control qubit as the data and the target qubit as the program, can successfully perform $U(\alpha)$ 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 $U(-\alpha)|\psi\rangle$. What we can now do is to send this qubit back into the same circuit, but with the program state that encodes the operation $U(2\alpha)$. 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 the other 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 $1-(1/2)^N$ [5].

In this paper 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.

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 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: First, one can measure and estimate the program state and with the classical information so obtained, one can *classically* control 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 negligibly 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 any information about the set of instructions. Another advantage of this approach is that the unknown program state can be efficiently *teleported* to a distant programmable quantum processor. Let us imagine a situation that on a board of a satellite we have a set of data encoded in a quantum state. Depending on the task we would like to perform different data processing (e.g., we want to perform operations or transformations that even might not be known at the time of the launch of the satellite). In this situation it might be very convenient to have a programmable processor on board the satellite. Then the set of (in principle, unknown) instructions encoded in the state of the program register can be teleported from a control center onto the satellite and a desired operation on the data can be performed.

II. OPERATIONS ON QUBITS

We shall begin by describing the methods developed in Refs. [4] and [5] in terms of the formalism presented in Ref. [6]. 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

$$G = \sum_{j,k=1}^{N} A_{jk} \otimes |j\rangle_p \, {}_p\langle k|, \tag{2.1}$$

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 [6]

$$\sum_{j=1}^{N} A_{jk_1}^{\dagger} A_{jk_2} = \sum_{j=1}^{N} A_{k_1 j} A_{k_2 j}^{\dagger} = I_d \delta_{k_1 k_2}, \tag{2.2}$$

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

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

where

$$A_j(\Xi) = \sum_{k=1}^{N} {}_p\langle k|\Xi\rangle_p A_{jk}. \tag{2.4}$$

Let us begin by applying this formalism to a CNOT gate, the simplest example considered in Ref. [5]. Both the data and program spaces are two dimensional, and the data space is the control qubit and the program space is the target qubit. Expressing the operator for the CNOT gate in the form given in Eq. (2.1) and choosing the basis $\{|0\rangle,|1\rangle\}$ for the program space, we find that

$$A_{00} = |0\rangle\langle 0|, \quad A_{01} = |1\rangle\langle 1|,$$

$$A_{10} = |1\rangle\langle 1|, \quad A_{11} = |0\rangle\langle 0|. \tag{2.5}$$

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

$$|\Xi(\alpha)\rangle = \frac{1}{\sqrt{2}}(e^{i\alpha}|0\rangle + e^{-i\alpha}|1\rangle). \tag{2.6}$$

This gives us the output state

$$G(|\psi\rangle_d \otimes |\Xi(\alpha)\rangle_p) = \sum_{j=0}^{1} A_j(\alpha)|\psi\rangle_d \otimes |j\rangle_p, \tag{2.7}$$

where the program operators are

$$A_0(\alpha) = \frac{e^{i\alpha}}{\sqrt{2}}|0\rangle\langle 0| + \frac{e^{-i\alpha}}{\sqrt{2}}|1\rangle\langle 1| = \frac{1}{\sqrt{2}}U(\alpha),$$

$$A_1(\alpha) = \frac{e^{i\alpha}}{\sqrt{2}}|1\rangle\langle 1| + \frac{e^{-i\alpha}}{\sqrt{2}}|0\rangle\langle 0| = \frac{1}{\sqrt{2}}U(-\alpha). \tag{2.8}$$

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

$$U(2\alpha)U(-\alpha)|\psi\rangle_d = U(\alpha)|\psi\rangle_d, \tag{2.9}$$

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 [5]. Qubit 1 is the data qubit, and qubits 2 and 3 are the program qubits. The first gate is a CNOT 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

$$|0\rangle_p = |0\rangle_2|0\rangle_3, \quad |2\rangle_p = |1\rangle_2|0\rangle_3,$$

$$|1\rangle_p = |0\rangle_2|1\rangle_3, \quad |3\rangle_p = |1\rangle_2|1\rangle_3, \tag{2.10}$$

then this circuit can be described by the operators

$$A_{00} = |0\rangle\langle 0|, \quad A_{01} = 0, \quad A_{02} = |1\rangle\langle 1|, \quad A_{03} = 0,$$

$$A_{10} = 0, \quad A_{11} = |0\rangle\langle 0|, \quad A_{12} = 0, \quad A_{13} = |1\rangle\langle 1|,$$

$$A_{20} = 0, \quad A_{21} = |1\rangle\langle 1|, \quad A_{22} = |0\rangle\langle 0|, \quad A_{23} = 0,$$

$$A_{30} = |1\rangle\langle 1|, \quad A_{31} = 0, \quad A_{32} = 0, \quad A_{33} = |0\rangle\langle 0|. \tag{2.11}$$

The program state is now

$$|\Xi(\alpha)\rangle = \frac{1}{2}\sum_{j=0}^{3} e^{i(3-2j)\alpha}|j\rangle_p. \tag{2.12}$$

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 CNOT gate with a single qubit program and consider a more general program state

$$|\Xi\rangle = c_0|0\rangle + c_1|1\rangle, \tag{2.13}$$

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

$$A_0(\Xi) = c_0|0\rangle\langle 0| + c_1|1\rangle\langle 1|,$$

$$A_1(\Xi) = c_1|0\rangle\langle 0| + c_0|1\rangle\langle 1|. \tag{2.14}$$

These operators are not unitary, but they do have the property that $A_0(\Xi)A_1(\Xi)=A_1(\Xi)A_0(\Xi)=c_0c_1I$. The output state of this circuit is given by Eq. (2.7), 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

$$B(z) = |0\rangle\langle 0| + z|1\rangle\langle 1|. \tag{2.15}$$

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

$$|\Xi'\rangle = \left(\frac{1}{1+|z|^4}\right)^{1/2}(|0\rangle + z^2|1\rangle). \tag{2.16}$$

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

$$A_{00} = |0\rangle\langle 0|, \quad A_{01} = |1\rangle\langle 1|, \quad A_{02} = 0, \quad A_{03} = 0,$$

$$A_{10} = 0, \quad A_{11} = |0\rangle\langle 0|, \quad A_{12} = |1\rangle\langle 1|, \quad A_{13} = 0,$$

$$A_{20} = 0, \quad A_{21} = 0, \quad A_{22} = |0\rangle\langle 0|, \quad A_{23} = |1\rangle\langle 1|,$$

$$A_{30} = |1\rangle\langle 1|, \quad A_{31} = 0, \quad A_{32} = 0, \quad A_{33} = |0\rangle\langle 0|. \tag{2.17}$$

The program state is now

$$|\Xi\rangle_p = \sum_{k=0}^{3} c_k|k\rangle_p, \tag{2.18}$$

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

$$|c_0|^2 = \frac{1-|z|^2}{1-|z|^8}. \tag{2.19}$$

The operation of the processor is given by

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = \sum_{j=0}^{3} A_j(\Xi)|\psi\rangle_d \otimes |j\rangle_p, \tag{2.20}$$

where

$$A_j(\Xi) = \sum_{k=0}^{3} c_k A_{jk}, \tag{2.21}$$

and the operators A_{jk} are given in Eq. (2.17). This processor will perform the operation $B(z)$ with a reasonably high probability. In order to see this, we first note that $A_j(\Xi)=z^jA_0(\Xi)$ for $j=0,1,2,$. This implies that

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = A_0(\Xi)|\psi\rangle_d \otimes \left(\sum_{j=0}^{2} z^j |j\rangle_p\right) + A_3(\Xi)|\psi\rangle_d \otimes |3\rangle_p, \qquad (2.22)$$

and $A_0(\Xi)=c_0B(z)$. 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

$$P_{suc} = \left(\frac{1-|z|^6}{1-|z|^8}\right)(|\alpha|^2 + |z|^2|\beta|^2). \qquad (2.23)$$

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

$$\overline{P}_{suc} = \frac{1}{2}\left(\frac{1-|z|^6}{1-|z|^8}\right)(1+|z|^2). \qquad (2.24)$$

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

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

$$A_{jk} = \delta_{j,k}|0\rangle\langle 0| + \delta_{j+1,k}|1\rangle\langle 1|, \qquad (2.25)$$

where the addition in the second Kronecker delta is done modulo N. These operators satisfy Eq. (2.2), so that they define a unitary operator. The program state is now

$$|\Xi\rangle = c_0 \sum_{j=0}^{N-1} z^j |j\rangle_p, \qquad (2.26)$$

where

$$|c_0|^2 = \frac{1-|z|^2}{1-|z|^{2N}}. \qquad (2.27)$$

This yields the following output state:

$$G(|\psi\rangle_d \otimes |\Xi\rangle_p) = c_0B(z)|\psi\rangle_d \otimes \sum_{j=0}^{N-2} z^j |j\rangle_p + A_{N-1}(\Xi)|\psi_d\rangle \otimes |N-1\rangle_p, \qquad (2.28)$$

where

$$A_{N-1}(\Xi) = c_0(z^{N-1}|0\rangle\langle 0| + |1\rangle\langle 1|). \qquad (2.29)$$

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

$$P_{suc} = 1 - \|A_{N-1}(\Xi)\psi\|^2 = 1 - \frac{(1-|z|^2)(|\alpha|^2|z|^{2(N-1)} + |\beta|^2)}{|z|^{2N}-1}. \qquad (2.30)$$

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

$$P_{suc} \to 1 - (1-|z|^2)|\beta|^2 = \|B(z)\psi\|^2, \qquad (2.31)$$

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

$$P_{suc} \to 1 - \left(1 - \frac{1}{|z|^2}\right)|\alpha|^2 = \frac{1}{|z|^2}\|B(z)\psi\|^2. \qquad (2.32)$$

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.

III. QUDITS

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}

$$A_{00} = |0\rangle\langle 0|, \quad A_{01} = |1\rangle\langle 1|, \quad A_{02} = |2\rangle\langle 2|,$$

$$A_{10} = |2\rangle\langle 2|, \quad A_{11} = |0\rangle\langle 0|, \quad A_{12} = |1\rangle\langle 1|,$$

$$A_{20} = |1\rangle\langle 1|, \quad A_{21} = |2\rangle\langle 2|, \quad A_{22} = |0\rangle\langle 0|. \qquad (3.1)$$

The general program state is

$$|\Xi\rangle = c_0|0\rangle + c_1|1\rangle + c_2|2\rangle, \qquad (3.2)$$

which gives the program operators

$$A_0(\Xi) = c_0|0\rangle\langle 0| + c_1|1\rangle\langle 1| + c_2|2\rangle\langle 2|,$$

$$A_1(\Xi) = c_0|2\rangle\langle 2| + c_1|0\rangle\langle 0| + c_2|1\rangle\langle 1|,$$

$$A_2(\Xi) = c_0|1\rangle\langle 1| + c_1|2\rangle\langle 2| + c_2|0\rangle\langle 0|. \qquad (3.3)$$

The output state is

$$|\Psi_{out}\rangle = \sum_{j=0}^{2} A_j(\Xi)|\psi\rangle_d \otimes |j\rangle_p, \qquad (3.4)$$

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

$$|\Xi'\rangle = c_0'|0\rangle + c_1'|1\rangle + c_2'|2\rangle. \qquad (3.5)$$

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

$$A_0(\Xi') = c_0'|0\rangle\langle 0| + c_1'|1\rangle\langle 1| + c_2'|2\rangle\langle 2|, \qquad (3.6)$$

we see that this condition is satisfied if

$$c_0' = \frac{\alpha c_0}{c_1}, \quad c_1' = \frac{\alpha c_1}{c_2}, \quad c_2' = \frac{\alpha c_2}{c_0}. \tag{3.7}$$

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 $\{|0\rangle_d,\dots,|D-1\rangle_d\}$. 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

$$B_0(z) = z|0\rangle_{p\,p}\langle 0| + X, \tag{3.8}$$

where

$$X = \sum_{k=1}^{D-1} |k\rangle_{p\,p}\langle k|. \tag{3.9}$$

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

$$A_{jk} = \delta_{jk}X + \delta_{k,j+1}|0\rangle_{p\,p}\langle 0|, \tag{3.10}$$

where all additions are modulo D. The program state

$$|\Xi\rangle_p = c_0\sum_{k=0}^{N-1} z^k|k\rangle_p, \tag{3.11}$$

where $|c_0|^2$ is given by Eq. (2.27), gives us for $0\leqslant j\leqslant N-2$

$$A_j(\Xi) = c_0 z^j B_0(z). \tag{3.12}$$

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

$$P_{suc} = \frac{|z|^{2(N-1)}-1}{|z|^{2N}-1}\|B_0\psi\|^2, \tag{3.13}$$

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 1 if $|z|=1$. If $|z|>1$ we have

$$P_{suc} \to \frac{1}{|z|^2}\|B_0\psi\|^2, \tag{3.14}$$

and if $|z|<1$, then

$$P_{suc} \to \|B_0\psi\|^2. \tag{3.15}$$

As before, we see that it is only in the case that the operation is unitary that the probability goes to 1.

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 1.

IV. REALIZATION OF SU(2) ROTATIONS

In the Vidal-Masanes-Cirac 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 CNOT 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 our earlier work [6] we have shown that 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. 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 CNOT gates, such that first 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 $|0\rangle|-\rangle,|1\rangle|+\rangle$ and $|1\rangle|-\rangle$.

The action of the QID processor is given by relation [10,6]

$$G = \sum_{j=0}^{3} \sigma_j \otimes |\Xi_j\rangle\langle\Xi_j|, \tag{4.1}$$

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.,

$$|\Xi_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle), |\Xi_x\rangle = \frac{1}{\sqrt{2}}(|01\rangle+|10\rangle),$$

$$|\Xi_z\rangle = \frac{1}{\sqrt{2}}(|00\rangle-|11\rangle), |\Xi_y\rangle = \frac{1}{\sqrt{2}}(|01\rangle-|10\rangle).$$

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

$$|\Xi(\vec{\mu})\rangle_p = \cos\,\mu|\Xi_0\rangle + i\frac{\sin\,\mu}{\mu}(\mu_x|\Xi_x\rangle + \mu_y|\Xi_y\rangle + \mu_z|\Xi_z\rangle). \tag{4.2}$$

Performing 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

$$U_{\vec{\mu}} = \cos\,\mu I + \frac{i\,\sin\,\mu}{\mu}(\mu_x\sigma_x + \mu_y\sigma_y + \mu_z\sigma_z). \tag{4.3}$$

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

$$|\psi\rangle_d \otimes |\Xi(\vec{\mu})\rangle_p \to \frac{1}{2}\left(\sum_{j=0}^{3}\sigma_j U_{\vec{\mu}}\sigma_j|\psi\rangle_d \otimes |\widetilde{j}\,\rangle_p\right), \tag{4.4}$$

where vectors $\{|\widetilde{j}\,\rangle_p\}$ form the basis of $\mathcal{H}_p$ associated with the realized measurement. The explicit form of the vectors is presented in following section 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 which 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 $p(1)=1/4+3/16=7/16$. Using more and more conditioned loops the success probability is given by $p(n) = \Sigma_{j=1}^{n}(3^{j-1}/4^j) = 1/4\Sigma_{j=0}(3/4)^j = 1-(3/4)^n$ which 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, $p \simeq 10^{-4}$.

The example of Vidal, Masanes, and Cirac shows us that we are able to replace the feedback scenario with a probabilistic scenario by using different processors. An open problem is whether the same replacement can be done in general, or at least for the case of the QID.

V. SU(N) ROTATIONS OF QUDITS

In what follow we will show that one can utilize the QID for a probabilistic implementation of SU(N) rotations of qudits. We start our discussion with a brief description of the QID in the case of qudits. First, we introduce a generalization of the two-qubit CNOT gate [10] for qudits. This is a conditional shift operator defined with a control qudit "*a*" and the target qudit "*b*",

$$D_{ab} = \sum_{k,m=0}^{N-1}|k\rangle_a\langle k| \otimes |(m+k)\text{mod } N\rangle_b\langle m|, \tag{5.1}$$

which implies that

$$D_{ab}^{\dagger} = \sum_{k,m=0}^{N-1}|k\rangle_a\langle k| \otimes |(m-k)\text{mod } N\rangle_b\langle m|. \tag{5.2}$$

From this definition it follows that the operator D_{ab} acts on the basis vectors of a qudit as

$$D_{ab}|k\rangle|m\rangle = |k\rangle|(k+m)\text{mod } N\rangle, \tag{5.3}$$

which means that this operator has the same action as the conditional adder and can be performed with the help of the simple quantum network discussed in Ref. [11]. Note that for $N>2$ the two operators D and $D^{\dagger}$ differ; they describe conditional shifts in opposite directions. Therefore the generalizations of the CNOT operator to higher dimensions are just conditional shifts.

Following our earlier work [6,10] we can assume the network for the probabilistic universal quantum processor to be

$$P_{123} = D_{31}D_{21}^{\dagger}D_{13}D_{12}. \tag{5.4}$$

The data register consists of system 1 and the program register of systems 2 and 3. The state $|\Xi_V\rangle_{23}$ acts as the "software" that carries the information about the operation V to be implemented on the qudit data state $|\Psi\rangle_1$. The output state of the three-qudit system, after the four controlled shifts are applied, reads

$$|\Omega\rangle_{123} = D_{31}D_{21}^{\dagger}D_{13}D_{12}|\Psi\rangle_1|\Xi_V\rangle_{23}. \tag{5.5}$$

The sequence of four operators acting on the basis vectors gives $|n\rangle_1|m\rangle_2|k\rangle_3$ as

$$D_{31}D_{21}^{\dagger}D_{13}D_{12}|n\rangle_1|m\rangle_2|k\rangle_3 = |(n-m+k)\text{mod } N\rangle_1|(m+n) \times\text{mod } N\rangle_2|(k+n)\text{mod } N\rangle_3. \tag{5.6}$$

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

$$|\Xi_{mn}\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\exp\left(i\frac{2\pi}{N}mk\right)|k\rangle|(k-n)\text{mod } N\rangle, \tag{5.7}$$

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

$$\begin{aligned}
P_{123}|\Psi\rangle_1|\Xi_{mn}\rangle_{23} &= \sum_{jk}\frac{\alpha_j}{\sqrt{N}}\exp\left[\frac{2\pi ikm}{N}\right]P_{123}|j\rangle|k\rangle|k-n\rangle \\
&= \sum_{jk}\frac{\alpha_j}{\sqrt{N}}\exp\frac{2\pi ikm}{N}|j-n\rangle|k+j\rangle|k+j-n\rangle \\
&= \sum_{jk}\alpha_j\exp\frac{-2\pi ijm}{N}|j-n\rangle|\Xi_{mn}\rangle \\
&= (U^{(m,n)}|\Psi\rangle)|\Xi_{mn}\rangle, \qquad (5.8)
\end{aligned}$$

where we have introduced the notation

$$U^{(m,n)} = \sum_{s=0}^{N-1}\exp\frac{-2i\pi sm}{N}|s-n\rangle\langle s|. \qquad (5.9)$$

This result is similar to the one we found in the case of a single qubit (see the preceding section). The operators $U^{(m,n)}$ satisfy the orthogonality relation

$$\mathrm{Tr}[(U^{(m',n')})^\dagger U^{(m,n)}] = N\delta_{m,m'}\delta_{n,n'}. \qquad (5.10)$$

The space of linear operators $\mathcal{T}(\mathcal{H})$ defined on some Hilbert space $\mathcal{H}$ with the scalar product given by Eq. (5.10) we know as *Hilbert-Schmidt space*. Thus the unitary operators $U^{(m,n)}$ form an orthogonal basis in it and any operator $V \in \mathcal{T}(\mathcal{H})$ can be expressed in terms of them,

$$V = \sum_{m,n=0}^{N-1} d_{mn}U^{(m,n)}. \qquad (5.11)$$

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

$$d_{mn} = \frac{1}{N}\mathrm{Tr}[(U^{(m,n)})^\dagger V]. \qquad (5.12)$$

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

$$|\Xi_V\rangle_{23} = \sum_{m,n=0}^{N-1} d_{mn}|\Xi_{mn}\rangle_{23}. \qquad (5.13)$$

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

$$|\Omega\rangle_{123} = \sum_{mn} d_{mn}U^{(m,n)}|\Psi\rangle_1 \otimes |\Xi_{mn}\rangle_{23}. \qquad (5.14)$$

Now let us perform a measurement of the program output in the basis

$$|\Phi_{rs}\rangle = \frac{1}{N}\sum_{m,n=0}^{N-1}\exp\left[2\pi i\frac{(mr-ns)}{N}\right]|\Xi_{mn}\rangle. \qquad (5.15)$$

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 $|\Phi_{rs}\rangle$ can be rewritten in a factorized form, i.e.,

$$|\Phi_{rs}\rangle = |-r\rangle \otimes \frac{1}{\sqrt{N}}\sum_{n=0}^{N}\exp\left[2\pi i\frac{ns}{N}\right]|n-r\rangle, \qquad (5.16)$$

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:

$$\begin{aligned}
P_{123}|\Psi\rangle_1|\Xi_V\rangle_{23} &= \sum_{m,n=0}^{N-1} d_{m,n}U^{(m,n)}|\Psi\rangle_1|\Xi_{mn}\rangle_{23} \\
&= \sum_{m,n=0}^{N-1} d_{m,n}U^{(m,n)}|\Psi\rangle_1 \\
&\quad\times\left[\frac{1}{N}\sum_{r,s=0}^{N-1}\exp\left[-2\pi i\frac{(mr-ns)}{N}\right]|\Phi_{rs}\rangle_{23}\right] \\
&= \frac{1}{N}\sum_{r,s=0}^{N-1}\sum_{m,n=0}^{N-1}\left\{\exp\left[-2\pi i\frac{(mr-ns)}{N}\right]\right. \\
&\quad\left.\times d_{m,n}U^{(m,n)}\right\}|\Psi\rangle_1|\Phi_{rs}\rangle_{23}. \qquad (5.17)
\end{aligned}$$

Taking into account that

$$[U^{(p,q)}]^\dagger U^{(m,n)}U^{(p,q)} = \exp\left[2\pi i\frac{(mq-np)}{N}\right]U^{(m,n)} \qquad (5.18)$$

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

$$\frac{1}{N}\mathrm{Tr}[(U^{(s,r)})^\dagger(U^{(m,n)})^\dagger U^{(s,r)}V] = \exp\left[-2\pi i\frac{(mr-ns)}{N}\right]d_{m,n}. \qquad (5.19)$$

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

$$P_{123}|\Psi\rangle_1|\Xi_V\rangle_{23} = \frac{1}{N}\sum_{r,s=0}^{N-1}[U^{(s,r)}V(U^{(s,r)})^\dagger]|\Psi\rangle_1|\Phi_{rs}\rangle_{23}, \qquad (5.20)$$

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 $[U^{(s,r)}V(U^{(s,r)})^\dagger]|\Psi\rangle_1$. 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

$$V^{(r,s)} = V[U^{(s,r)}V(U^{(s,r)})^{\dagger}]^{-1}. \tag{5.21}$$

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 is $p(K)=1-(1-1/N^2)^K$.

VI. CONCLUSIONS

In this paper we have analyzed a probabilistic programmable quantum processor. 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. In our paper we have analyzed systematically how to perform U(1) rotations of qubits and qudits and one-parameter families of nonunitary operations, where the parameter 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. It is also possible to use enlarged programs to increase the probability of success without the use of loops. 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.

ACKNOWLEDGMENTS

This work was supported in part by the European Union projects QGATES and CONQUEST, by the National Science Foundation under Grant No. PHY-0139692, and by the Slovak Academy of Sciences.

[1] M. Nielsen and I. L. Chuang, Phys. Rev. Lett. **79**, 321 (1997).
[2] M. Hillery, V. Bužek, and M. Ziman, Fortschr. Phys. **49**, 987 (2001).
[3] M. Hillery, M. Ziman, and V. Bužek, Phys. Rev. A **66**, 042302 (2002).
[4] J. Preskill, Proc. R. Soc. London, Ser. A **454**, 385 (1998).
[5] G. Vidal, L. Masanes, and J. I. Cirac, Phys. Rev. Lett. **88**, 047905 (2002).
[6] M. Hillery, V. Bužek, and M. Ziman, Phys. Rev. A **65**, 022301 (2002).
[7] M. Dušek and V. Bužek, Phys. Rev. A **66**, 022112 (2002).
[8] A. K. Ekert, C. M. Alves, D. K. L. Oi, M. Horodecki, P. Horodecki, and L. C. Kwek, Phys. Rev. Lett. **88**, 217901 (2002).
[9] J. P. Paz and A. Roncaglia, Phys. Rev. A **68**, 052316 (2003).
[10] S. L. Braunstein, V. Bužek, and M. Hillery, Phys. Rev. A **63**, 052313 (2001).
[11] V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).