Approximate programmable quantum processors

Mark Hillery,[1,2] Mário Ziman,[2,3] and Vladimír Bužek[2,4]
[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, 845 11 Bratislava, Slovakia*
[3]*Quniverse, Líščie údolie 116, 841 04 Bratislava, Slovakia*
[4]*Abteilung für Quantenphysik, Universität Ulm, 89069 Ulm, Germany*

(Received 13 October 2005; published 24 February 2006)

A quantum processor is a programmable quantum circuit in which both the data and the program, which specifies the operation that is carried out on the data, are quantum states. We study the situation in which we want to use such a processor to approximate a set of unitary operators to a specified level of precision. We measure how well an operation is performed by the process fidelity between the desired operation and the operation produced by the processor. We show how to find the program for a given processor that produces the best approximation of a particular unitary operation. We also place bounds on the dimension of the program space that is necessary to approximate a set of unitary operators to a specified level of precision.

DOI: 10.1103/PhysRevA.73.022345 PACS number(s): 03.67.Lx, 03.65.Ta

I. INTRODUCTION

Quantum circuits are typically designed to perform one function, for example, teleportation or cloning. It is useful to have circuits that are more flexible and can perform a variety of functions. The operation that the circuit performs can either be determined by classically setting the values of some parameters, for example the rotation angle in a one-qubit rotation gate, or it can be determined quantum mechanically, where a quantum system serves as a program to tell the circuit what to do. The second method has the advantage that the program could be a result of a previous stage of a quantum computation, which would allow one stage of a computation to control a subsequent one.

A programmable quantum circuit (quantum processor) has two inputs, the data register and the program register. The data register is in the state on which we want to perform an operation, and the program state specifies the operation. An example is the controlled-NOT (CNOT) gate in which the control qubit is the program and the target qubit is the data. If the control qubit is in the state $|0\rangle$, nothing is done to the data state, and if it is in the state $|1\rangle$, the operation σ_x (bit flip) is applied to the data state. The superposition state $\alpha|0\rangle+\beta|1\rangle$ causes the completely positive quantum map

$$T(\rho)=|\alpha|^2\rho+|\beta|^2\sigma_x\rho\sigma_x \tag{1.1}$$

to be applied to the data state $\rho=|\psi\rangle\langle\psi|$.

Suppose that we have a set of N unitary operators that we want to be able to implement on the data qubit with a programmable quantum circuit. Nielsen and Chuang showed that this requires a program space of at least N dimensions [1]. This follows from the fact that the program states corresponding to any two of the unitary operators must be orthogonal. If one wants to be able to realize a large (or infinite) number of unitary operations with a program space of fixed dimension, one has two possible options. One option is to make the processor probabilistic, that is, a measurement is performed at the program output, and if the correct result is obtained, the desired operation has been performed on the data [1–5]. The probability of obtaining the proper measurement outcome will, in general, be less than 1 so that the processor succeeds with only a certain probability. The second option is to make the processor an approximate one. That is, each of the operations is not performed exactly, but only up to some level of approximation. It is this type of processor that we wish to discuss here.

Approximate processors have been discussed by Vlasov [2] and by Vidal and Cirac [3]. Vlasov considered a classically programmable processor, while Vidal and Cirac considered one whose programs are arbitrary quantum states. They made some rough estimates of the resources required for a processor to be able to program a set of unitary operators to a specified level of precision [3]. In Refs. [6–8] approximate programmable quantum measurement devices have been studied. These devices realize certain classes of positive operator-valued measures (POVM's) up to some level of approximation, and which POVM they perform is determined by a program state. It was shown by D'Ariano and Perinotti that for programmable measurement devices the number of dimensions of the program space is a polynomial function of the reciprocal of the desired accuracy [8].

Physical limitations in real systems lead to an additional reason to study approximate processors. That is, ideal devices in theory become approximate ones in practice. For example, if one wants to perform a rotation on a qubit that is a two-level atom, one applies a classical field to the atom. Real fields, however, consist of photons, and this and energy constraints on the field place limits on the accuracy of the rotation that can be achieved [9–12]. Conservation laws can also place limits on the accuracy of quantum operations [13].

Other types of processors have been explored. In particular, processors that evaluate the expectation value of an arbitrary operator have been proposed [14,15]. In these processors, the data are the state in which the expectation value is to be evaluated, and the program specifies the operator.

In this paper 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.

II. OPTIMAL PROGRAM STATES

We now 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 [16,17]

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

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

$$G=\sum_{j,k=1}^{N}A_{jk}\otimes|j\rangle_{pp}\langle k|. \tag{2.2}$$

These operators obey the relations

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

$$\sum_{k=1}^{N}A_{j_1k}^{\dagger}A_{j_2k}=I_d\delta_{j_1j_2}, \tag{2.3}$$

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

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

from which it follows that

$$\sum_{j=1}^{N}A_j^{\dagger}(\Xi)A_j(\Xi)=I_d. \tag{2.5}$$

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 [18], which was originally proposed by Raginsky [19]. 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

$$|\Phi\rangle=\frac{1}{\sqrt{D}}\sum_{j=1}^{D}|j\rangle|j\rangle, \tag{2.6}$$

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

$$\rho_j=(\mathcal{I}\otimes T_j)(|\Phi\rangle\langle\Phi|), \tag{2.7}$$

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

$$F_{proc}(T_1,T_2)=(\mathrm{Tr}\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}})^2. \tag{2.8}$$

The process fidelity has a number of useful properties that are discussed in Refs. [19] and [18], one of which is the fact that it is symmetric, i.e., $F_{proc}(T_1,T_2)=F_{proc}(T_2,T_1)$.

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

$$\mathrm{Tr}\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}=\frac{1}{D}\left(\sum_{j_1,j_2=1}^{D}\langle j_1|U^{-1}T_2(|j_1\rangle\langle j_2|)U|j_2\rangle\right)^{1/2}. \tag{2.9}$$

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

$$T_2(\rho)=\sum_{j=1}^{N}A_j(\Xi)\rho_dA_j(\Xi)^{\dagger}, \tag{2.10}$$

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

$$F(U,T_2)=\frac{1}{D^2}\sum_{j=1}^{N}|\mathrm{Tr}[U^{-1}A_j(\Xi)]|^2. \tag{2.11}$$

Using the notation for the Hilbert-Schmidt scalar product $(A|B)=\mathrm{Tr}\,A^{\dagger}B$ this can be rewritten in the form $F(U,T_2)=(1/D^2)\Sigma_j|(A_j(\Xi)|U)|^2$. A similar result was derived independently by D'Ariano and Perinotti in [20].

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

$$M_{k_1k_2}=\frac{1}{D^2}\sum_{j=1}^{N}\mathrm{Tr}(A_{jk_1}^{\dagger}U)\mathrm{Tr}(U^{-1}A_{jk_2}), \tag{2.12}$$

we have, from Eq. (2.4), that

$$F(U,T_2)=\sum_{k_1,k_2=1}^{N}{}_p\langle\Xi|k_1\rangle_pM_{k_1k_2p}\langle k_2|\Xi\rangle_p. \tag{2.13}$$

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. (2.13) shows that this can be accomplished by finding the eigenvector of $M=\Sigma_{k_1,k_2}M_{k_1k_2}|k_1\rangle\langle k_2|$ 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. [16] 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

$$M_{k_1k_2}=\frac{1}{D^2}|\mathrm{Tr}(U^\dagger U_{k_1})|^2\delta_{k_1k_2}. \tag{2.14}$$

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.

III. 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

$$|\theta\rangle=\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}e^{-ik\theta}|k\rangle. \tag{3.1}$$

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

$$E_+|\theta_m\rangle=e^{i\theta_m}|\theta_m\rangle,\quad E_-|\theta_m\rangle=e^{-i\theta_m}|\theta_m\rangle. \tag{3.2}$$

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 $\mathcal{H}_d\otimes\mathcal{H}_p$,

$$G=\exp\left[i\left(\frac{\pi}{2}\right)(\sigma^{(+)}\otimes E_-+\sigma^{(-)}\otimes E_+)\right]. \tag{3.3}$$

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

$$\begin{aligned}|\Omega_m\rangle&=G(|\psi\rangle_d\otimes|\theta_m\rangle_p)\\&=\exp\left[i\left(\frac{\pi}{2}\right)(e^{-i\theta_m}\sigma^{(+)}+e^{i\theta_m}\sigma^{(-)})\right]|\psi\rangle_d\otimes|\theta_m\rangle_p.\end{aligned} \tag{3.4}$$

Defining

$$U(\theta)=\exp\left[i\left(\frac{\pi}{2}\right)(e^{-i\theta}\sigma^{(+)}+e^{i\theta}\sigma^{(-)})\right], \tag{3.5}$$

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, that 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

$$F=\frac{1}{N^2}\sum_{m=0}^{N-1}\cos^2(\theta_m-\theta)\frac{\sin^2[N(\theta_m-\theta)/2]}{\sin^2[(\theta_m-\theta)/2]}. \tag{3.6}$$

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

$$F(U(\theta_1),U(\theta_2))=\cos^2(\theta_1-\theta_2). \tag{3.7}$$

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

$$F\geqslant\cos^2\left(\frac{\pi}{N}\right)\sim 1-\left(\frac{\pi}{N}\right)^2. \tag{3.8}$$

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 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.

IV. BOUND ON DIMENSION OF PROGRAM SPACE

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 $|(A|B)|\leqslant\sqrt{(A|A)(B|B)}$ implies that

$$|\mathrm{Tr}[U^\dagger A_j(\Xi)]|\leqslant\sqrt{D}\{\mathrm{Tr}[A_j^\dagger(\Xi)A_j(\Xi)]\}^{1/2}, \tag{4.1}$$

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

$$F(U,T)=\frac{1}{D^2}\sum_{j=1}^{N}|\mathrm{Tr}[U^{\dagger}A_j(\Xi)]|^2 \leqslant \frac{1}{D}\sum_{j=1}^{N}\mathrm{Tr}[A_j^{\dagger}(\Xi)A_j(\Xi)]=1. \tag{4.2}$$

In the last equality we used the normalization property of Kraus operators (2.5), i.e., $\Sigma_j A_j^{\dagger}(\Xi)A_j(\Xi)=I$.

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 the Schwartz inequality has to be saturated. This means that the operators $A_j(\Xi)$ and U are collinear, i.e., $A_j(\Xi)=\beta_j U$, where β_j is a complex number. Furthermore, Eq. (2.5) implies $\Sigma_{j=1}^{N}|\beta_j|^2=1$. 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

$$\sum_{j=1}^{N}\beta_{1j}^{*}\beta_{2j}U_1^{-1}U_2=\sum_{j=1}^{N}A_j^{\dagger}(\Xi_1)A_j(\Xi_2)=I_{dp}\langle\Xi_1|\Xi_2\rangle_p, \tag{4.3}$$

where we have used Eqs. (2.4) and (2.3). If $U_1\neq U_2$, then this equation implies that both ${}_p\langle\Xi_1|\Xi_2\rangle_p$ and $\Sigma_{j=1}^{N}\beta_{1j}^{*}\beta_{2j}$ 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., $F(U,T)\geqslant 1-\epsilon$, where T is specified by Kraus operators $A_j(\Xi)$. Let us express these operators as

$$A_j(\Xi)=\beta_j U+B_j(\Xi), \tag{4.4}$$

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

$$1\geqslant F(U,T)=\frac{1}{D^2}\sum_{j=1}^{N}|(U|A_j(\Xi))|^2=\sum_{j=1}^{N}|\beta_j|^2\geqslant 1-\epsilon. \tag{4.5}$$

Tracing both sides of the normalization condition

$$\sum\nolimits_j A_j(\Xi)^{\dagger}A_j(\Xi)=I$$

we obtain the inequality

$$\sum\nolimits_j \mathrm{Tr}[B_j(\Xi)^{\dagger}B_j(\Xi)]=\sum\nolimits_j (B_j(\Xi)|B_j(\Xi))\leqslant D\epsilon$$

.

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

$$A_j(\Xi_1)=\beta_{1j}U_1+B_{1j}(\Xi_1),$$
$$A_j(\Xi_2)=\beta_{2j}U_2+B_{2j}(\Xi_2), \tag{4.6}$$

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

$$I_d\langle\Xi_1|\Xi_2\rangle=\sum_{j=1}^{N}A_j(\Xi_1)^{\dagger}A_j(\Xi_2) =\sum_{j=1}^{N}[\beta_{1j}^{*}U_1^{\dagger}+B_{1j}^{\dagger}(\Xi_1)][\beta_{2j}U_2+B_{2j}(\Xi_2)]. \tag{4.7}$$

Let us evaluate the absolute value of the traces of both sides

$$D|\langle\Xi_1|\Xi_2\rangle|=\left|\sum_j (A_j(\Xi_1)|A_j(\Xi_2))\right|=\left|\sum_j [\beta_{1j}^{*}\beta_{2j}(U_1|U_2)+\beta_{1j}^{*}(U_1|B_{2j})+\beta_{2j}(B_{1j}|U_2)+(B_{1j}|B_{2j})]\right| \leqslant |(U_1|U_2)|\cdot\left|\sum_j \beta_{1j}^{*}\beta_{2j}\right|+2D\sqrt{\epsilon}+D\epsilon. \tag{4.8}$$

In the last line we used the formulas

$$\sum_j |(B_{1j}|B_{2j})|\leqslant\sum_j\sqrt{(B_{1j}|B_{1j})(B_{2j}|B_{2j})} \leqslant\sqrt{\sum_j (B_{1j}|B_{1j})\sum_j (B_{2j}|B_{2j})}\leqslant D\epsilon, \tag{4.9}$$

and

$$\left|\sum_j \beta_{1j}^{*}(U_1|B_{2j})\right|\leqslant\sum_j|\beta_{1j}|\sqrt{(U_1|U_1)(B_{2j}|B_{2j})}\leqslant D\sqrt{\epsilon}. \tag{4.10}$$

As a result we obtain the bound on the inner product between two program states,

$$|\langle\Xi_1|\Xi_2\rangle|\leqslant\frac{1}{D}|(U_1|U_2)|\left|\sum_j \beta_{1j}^{*}\beta_{2j}\right|+2\sqrt{\epsilon}+\epsilon. \tag{4.11}$$

Next we will estimate the first term. The idea is to use Eq. (4.7) 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.,

$$\eta=\max_{\psi}(1-|\langle\psi|U_1^{\dagger}U_2|\psi\rangle|^2). \tag{4.12}$$

This quantity describes the distinguishability of two unitary transformations, and a short calculation shows that $\eta\leqslant\|U_1-U_2\|^2$. After applying both sides of Eq. (4.7) 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

$$0 = P_{\eta}^{\perp}\left[\sum_{j} A_j(\Xi_1)^{\dagger}A_j(\Xi_2)\right]|\psi_{\eta}\rangle = \sum_{j} \beta_{1j}^{*}\beta_{2j}P_{\eta}^{\perp}U_1^{\dagger}U_2|\psi_{\eta}\rangle + |\omega\rangle, \tag{4.13}$$

where

$$|\omega\rangle = P_{\eta}^{\perp}\sum_{j=1}^{N}[\beta_{1j}^{*}U_1^{\dagger}B_{2j}(\Xi_2) + \beta_{2j}B_{1j}^{\dagger}(\Xi_1)U_2 + B_{1j}^{\dagger}(\Xi_1)B_{2j}(\Xi_2)]|\psi_{\eta}\rangle. \tag{4.14}$$

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

$$\left\|\sum_{j=1}^{N} B_j^{\dagger}(\Xi_1)B_j(\Xi_2)\right\| \le \sum_{j=1}^{N} (B_j(\Xi_1)|B_j(\Xi_1))^{1/2} \times (B_j(\Xi_2)|B_j(\Xi_2))^{1/2} \le \epsilon D \tag{4.15}$$

and

$$\left\|\sum_{j=1}^{N} \beta_{1j}^{*}U_1^{\dagger}B_{2j}(\Xi_2)\right\| \le \sum_{j=1}^{N} |\beta_{1j}|(B_j(\Xi_2)|B_j(\Xi_2))^{1/2} \le \sqrt{\epsilon D}. \tag{4.16}$$

Applying these inequalities we have that $\|\omega\| \le \epsilon D + 2\sqrt{\epsilon D}$. In addition, we find that $\|P_{\eta}^{\perp}U_1^{\dagger}U_2\psi_{\eta}\| = \sqrt{\eta}$. Therefore, we can conclude

$$\left|\sum_{j=1}^{N} \beta_{1j}^{*}\beta_{2j}\right| \le \frac{\epsilon D + 2\sqrt{\epsilon D}}{\sqrt{\eta}}. \tag{4.17}$$

Defining

$$F = \min\left(1, \frac{\epsilon D + 2\sqrt{\epsilon D}}{\eta}\right), \tag{4.18}$$

we have, finally, that

$$|\langle\Xi_1|\Xi_2\rangle| \le \frac{F}{D}|(U_1|U_2)| + 2\sqrt{\epsilon} + \epsilon. \tag{4.19}$$

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 [3,16]. 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

$$|\langle\Xi_j|\Xi_k\rangle| \le Y_{max} + 2\sqrt{\epsilon} + \epsilon = q(Y_{max}, \epsilon). \tag{4.20}$$

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 $M' \le K_q$, 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

$$q(0, \epsilon) = 2\sqrt{2\epsilon} + \epsilon. \tag{4.21}$$

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.

V. 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 [3] a probabilistic processor without measurement can be used as an approximate processor. In this case the transformation can be expressed as

$$\mathcal{E}_{\xi}[\varrho] = p_{success}\mathcal{T}[\varrho] + p_{error}\mathcal{N}[\varrho], \tag{5.1}$$

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 $p_{success} \le F(\mathcal{E}_{\xi}, \mathcal{T})$, 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

$$U(\theta) = e^{i\theta}|1\rangle\langle 1| + X, \tag{5.2}$$

where $X = \sum_{k=2}^{D}|k\rangle\langle k|$, and $0 \le \theta < 2\pi$.

Consider the processor described by the operators A_{jk} for $1 \le j, k \le N$, where

$$A_{jk} = \begin{cases} \delta_{jk}X + \delta_{k,j+1}|1\rangle\langle 1|, & j < N, \\ \delta_{Nk}X + \delta_{k,1}|1\rangle\langle 1|, & j = N, \end{cases} \tag{5.3}$$

originally described in Ref. [5]. With the program state

$$|\Xi\rangle = \frac{1}{\sqrt{N}}\sum_{k=1}^{N} e^{i(k-1)\theta}|k\rangle, \tag{5.4}$$

we find that for $1 \leqslant j \leqslant N-1$

$$A_j(\Xi) = \frac{1}{\sqrt{N}} e^{i(j-1)\theta} U(\theta), \tag{5.5}$$

and for $j = N$

$$A_N(\Xi) = \frac{1}{\sqrt{N}}(e^{i(N-1)\theta}X + |1\rangle\langle 1|). \tag{5.6}$$

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

$$F = 1 - \frac{2(D-1)}{ND^2}[1 - \cos(N\theta)]. \tag{5.7}$$

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 $2j\Delta\theta \leqslant \theta \leqslant 2(j+1)\Delta\theta$ by $U_j = U(\theta_j) = U((2j+1)\Delta\theta)$, where $j = 0, 1, \ldots, N-1$. We now define a U processor by setting, for $j,k = 0, 1, \ldots, N-1$,

$$A_{jk} = \delta_{jk}U_j. \tag{5.8}$$

In order to approximate $U(\theta)$ for $2j\Delta\theta \leqslant \theta \leqslant 2(j+1)\Delta\theta$, we choose the program state $|\Xi\rangle_p = |j\rangle_p$. For this processor we find that

$$\begin{aligned} 1 - F = \min_j \frac{2(D-1)}{D^2}[1 - \cos(\theta - \theta_j)] \leqslant \\ \frac{2(D-1)}{D^2}(1 - \cos\Delta\theta) \\ \sim \frac{2(D-1)}{D^2}\frac{\pi^2}{4N^2}. \end{aligned} \tag{5.9}$$

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.

This result shows that simply using a probabilistic processor, without changing the set of program states, as an approximate processor does not provide the best accuracy for a specified program space dimension. However, it is important to note that if we change the program states for the probabilistic processor when using it as an approximate one, better precision can be obtained. Let us return to our example, and consider the probabilistic processor with a general program state $|\Xi\rangle$. With this program, we find that the process fidelity between $U(\theta)$ and the operation implemented by the processor is

$$F = 1 - \frac{2(D-1)}{D^2}[1 - \mathrm{Re}(e^{i\theta}\langle\Xi|U_{shift}|\Xi\rangle]), \tag{5.10}$$

where U_{shift} is the shift operator $U_{shift}|k\rangle = |k+1\rangle$, where the addition is modulo D. Let us take as program states the eigenstates of U_{shift}, and note that the eigenvalues of U_{shift} are given by $e^{i\varphi}$, where $\varphi = 2(k+1)\pi/N$ for $k = 0, \ldots, N-1$. Choosing one of these states as our program state, we find that the fidelity is given by $F = 1 - [2(D-1)/D^2][1 - \cos(\theta + \varphi)]$. This is very similar to the result we obtained from the processor specified by Eq. (5.8). In choosing the program state to best approximate the operator $U(\theta)$, we select the eigenstate of U_{shift} for which $\cos(\varphi + \theta)$ is maximal. Doing so we obtain the same bound on the accuracy as in Eq. (5.9). Consequently, we find that our two processors have the same fidelity and approximate the operators $U(\theta)$ for $0 \leqslant \theta < 2\pi$ with the same accuracy. What this shows is that in using a probabilistic processor as an approximate one, it is sometimes possible to obtain better results by changing the set of program states that are used. The program states that are useful for operating the processor in a probabilistic mode, where the goal is to realize an operator perfectly some of the time, are not always the best ones for using it in a deterministic one, where the goal is to accurately approximate an operator every time.

VI. CONCLUSION

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 $\epsilon_G = 1 - \min_{\mathcal{E}\in\Gamma}\max_{\xi\in\mathcal{S}(\mathcal{H}_p)}F(\mathcal{E}, \mathcal{E}_\xi)$ [21]. 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 1 (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. [20] 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 [22], i.e., $G_{\mathrm{SWAP}}(|\psi\rangle\otimes|\phi\rangle)=|\phi\rangle\otimes|\psi\rangle$ for all states $|\psi\rangle,|\phi\rangle$. For a processor with both the data and program spaces having the same dimension D and G 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 $\varepsilon_G=1-1/D^2$. We suspect that this is the optimal value if the size of the program register equals 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 [22].

There are many open issues remaining. One possibility is to shift our focus, and rather than ask what type of 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 1. 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 probabilitic processor can be used as an approximate one.

ACKNOWLEDGMENTS

This work was supported in part by the European Union projects QGATES, QAP, and CONQUEST, by the Slovak Academy of Sciences via the project CE-PI under the Contract No. I/2/2005, and by the project APVT-99-012304. V.B. thanks the Alexander von Humboldt Foundation for support.

[1] M. A. Nielsen and I. L. Chuang, Phys. Rev. Lett. **79**, 321 (1997).
[2] A. Yu. Vlasov, e-print quant-ph/0103119.
[3] G. Vidal and J. I. Cirac, e-print quant-ph/0012067.
[4] M. Hillery, V. Bužek, and M. Ziman, Phys. Rev. A **65**, 022301 (2002).
[5] M. Hillery, M. Ziman, and V. Bužek, Phys. Rev. A **69**, 042311 (2004).
[6] M. Dušek and V. Bužek, Phys. Rev. A **66**, 022112 (2002).
[7] J. Fiurašek, M. Dušek, and R. Filip, Phys. Rev. Lett. **89**, 190401 (2002).
[8] G. M. D'Ariano and P. Perinotti, Phys. Rev. Lett. **94**, 090401 (2005).
[9] J. Gea-Banacloche, Phys. Rev. A **65**, 022308 (2002).
[10] A. Silberfarb and I. H. Deutsch, Phys. Rev. A **68**, 013817 (2003).
[11] W. M. Itano, Phys. Rev. A **68**, 046301 (2003).
[12] J. Gea-Banacloche, Phys. Rev. A **68**, 046303 (2002).
[13] M. Ozawa, Phys. Rev. Lett. **89**, 057902 (2002).
[14] A. K. Ekert, C. M. Alves, D. K. L. Oi, M. Horodecki, P. Horodecki, and L. C. Kwek, Phys. Rev. Lett. **88**, 217901 (2002).
[15] J. P. Paz and A. Roncaglia, Phys. Rev. A **68**, 052316 (2003).
[16] M. Hillery, M. Ziman, and V. Bužek, Phys. Rev. A **66**, 042302 (2002).
[17] M. Ziman and V. Bužek, Int. J. Quantum Inf. **1**, 523 (2003).
[18] A. Gilchrist, N. K. Langford, and M. A. Nielsen, Phys. Rev. A **71**, 062310 (2005).
[19] M. Raginsky, Phys. Lett. A **290**, 11 (2001).
[20] G. M. D'Ariano and P. Perinotti, e-print quant-ph/0509183; e-print quant-ph/0510033.
[21] 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. The complete bounded (CB) norm is defined as $\|E\|_{\mathrm{cb}}=\|E\otimes I_n\|$ where the symbol $\|\cdot\|$ denotes the usual operator norm (see, for instance Ref. 19).
[22] M. Hillery, V. Bužek, and M. Ziman (unpublished).