Eur. Phys. J. D **37**, 275–281 (2006)
DOI: 10.1140/epjd/e2005-00286-2

THE EUROPEAN PHYSICAL JOURNAL D

Simulation of generators of Markovian dynamics on programmable quantum processors

M. Koniorczyk[1,2,a], V. Bužek[1,3], and P. Adam[2,4]

[1] Research Center for Quantum Information, Slovak Academy of Sciences Dúbravská cesta 9, 845 11 Bratislava, Slovakia
[2] Department of Nonlinear and Quantum Optics, Research Institute for Solid State Physics and Optics, Hungarian Academy of Sciences, P.O. Box 49, 1525 Budapest, Hungary
[3] *Quniverse*, Líščie údolie 116, 841 04 Bratislava, Slovakia
[4] Institute of Physics, University of Pécs, Ifjúság út 6., 7624 Pécs, Hungary

Received 4 April 2005 / Received in final form 9 August 2005
Published online 25 October 2005 –

Abstract. We study how generators of Markovian dynamics of a qubit can be simulated using a programmable quantum processor.

PACS. 03.67.Lx Quantum computation – 03.65.Yz Decoherence; open systems; quantum statistical methods

1 Introduction

Quantum computing offers a new perspective in simulating physical systems [1,2]. While the simulation of a large quantum system is intractable for a classical computer, it should be possible of course on a quantum computer. Quantum computers are, from a theoretical point of view, highly "simplified" quantum systems. A simulation of a real physical system using such arrangement is interesting not only because of the usefulness of the simulation result, but also because it may reveal fundamental aspects of the real physical system, which may be obscure otherwise.

Maybe the most general problem of this kind would be the implementation of general quantum operations [3], the most general dynamics (described by completely positive — CP — maps) a quantum system can undergo. Consider a quantum system with a finite dimensional Hilbert space. As a consequence of Stinespring theorem [4], one can implement any completely positive map acting on this system by performing a unitary operation on the original system that is supplemented by an ancillary system. The unitary operation and the initial state of the ancillary system induce a desired CP map on the original system. This implementation of CP maps can be regarded as a "simulation" of quantum dynamics that is controlled by an initial state of the ancilla and the choice of the unitary operation acting on the ancilla and the system under consideration. Having in mind implementation of CP maps via unitary operations on enlarged systems it is natural to ask a question: if we assume a specific operation U, which maps can be induced on the system when we consider different initial states of the ancillary system?

The idea of programmable quantum gate arrays or quantum processors provide a specific approach to this general problem. The quantum processor has two input registers, one storing the state of a quantum system subjected to the operation (the "data register"), while the other one, the "program register", contains the description of the operation to be performed. After the operation of the circuit, the remains of the program state in the program register may either be omitted (deterministic regime) or be subjected to a measurement (probabilistic regime). Nielsen and Chuang [5] have shown that in the deterministic regime every implementable unitary operation requires an extra Hilbert space dimension in the program register. Thus having finite resources, a finite number of unitaries can be implemented this way. A controlled U gate is a prototype of such a processor: it implements the identity operator and one possible unitary. It was shown by Hillery et al. [6,7] that in the probabilistic case any operation can be implemented, though some of them with quite low probability of success. Vidal et al. [8] have presented a probabilistic scheme implementing unitary transformations with rather high probability.

In this paper we restrict ourselves to the deterministic regime, that is, the remains of the program register are dropped. Hillery et al. [9] have studied the possibility of implementing general quantum maps in this way, and have found certain limitations, e.g., an amplitude damping channel cannot be implemented. We focus here on a specific subset of quantum operations, namely Markovian dynamics. These processes are the most important ones in

[a] e-mail: kmatyas@szfki.hu

the description of quantum decoherence. The question of simulating Markovian dynamics in the quantum computing context was investigated by Bacon et al. [10]. As one of their main results, these authors provide a decomposition rule to build more complex dynamics from simpler "primitive" operations.

Recently the collision-like models of decoherence were worked out in the context of quantum information processing [11–15]. These models are microscopic, and thus they provide the opportunity of elementary understanding of noise and decoherence processes, and facilitate the description of entanglement dynamics in the system. They can be potentially generalized to more complicated systems, non-Markovian processes, etc. The main motivation behind these studies is to attain a better physical understanding of noise and decoherence processes. The present contribution is also in this spirit.

We describe a quantum logic network approach, related to programmable quantum circuits, for the simulation of decoherence. We intend to simulate an elementary time step of Markovian dynamics on a programmable quantum logic array. Otherwise speaking, the effect of the generator of the evolution is to be simulated. The generator characterizes the whole time evolution in the Markovian case.

We consider a collision type scheme where the single run of the processor implements an *infinitesimal time step* of the evolution approximately. We require that the actual length of this infinitesimal time step should be encoded into the initial state of the program register. In addition to this, some other parameters of the evolution can be transferred to the program state too. As the quantum circuit we consider a controlled U gate, and also quantum teleportation [16] as a programmable quantum circuit realizing a kind of closed loop control, which was already studied in the context of decoherence [17].

The paper is organized as follows: in Section 2 we review some definitions concerning Markovian dynamics, introducing Liouvillian superoperators, the generators of the dynamics. In Section 3 we describe the general idea of simulating a generator, as understood in this paper. Section 4 describes a reversible scheme capable of simulating a phase damping channel about an arbitrary axis. A geometrical interpretation of the result, and a comparison to another collision type scheme is also given. In Section 5 we discuss an application of Bennett's teleportation scheme in this context. In Section 6 the results are summarized and conclusions are drawn.

2 Markovian dynamics of a qubit

In this Section we review the definition of Markovian semigroup and its generators very briefly [18]. The most general operation, that a state of a qubit can undergo is described by a completely positive linear, trace preserving map acting on the set of density operators, also called a superoperator. This may be written in the Krauss-representation [3] as

$$\mathcal{E}(\varrho) = \sum_k E_k \varrho E_k^\dagger \tag{1}$$

where the E_k-s are positive operators such that $\sum E_k^\dagger E_k = 1$.

In order to introduce Markovian processes, one equips the set of superoperators with a continuous parameter t which is called the time. Stationary and Markovian dynamics obey the property

$$\mathcal{E}_{t_1}\mathcal{E}_{t_2} = \mathcal{E}_{t_1+t_2}, \tag{2}$$

with $t_1, t_2 > 0$. The set of superoperators with property (2) form a one-parameter semigroup, the Markovian semigroup. We also require the property

$$\mathcal{E}_0 = \hat{1}, \tag{3}$$

(where $\hat{1}$ stands for the identity superoperator) to hold.

The property in equation (2) enables us to define the generators of the semigroup as

$$\hat{L}(\varrho) = \lim_{t\to 0+0} \frac{\mathcal{E}_t(\varrho) - \varrho}{t}. \tag{4}$$

The operator $\hat{L}$ is the infinitesimal generator of the time evolution:

$$\mathcal{E}_t = \exp(\hat{L}t) = \lim_{n\to\infty} \left(\hat{1} - \frac{t}{n}\hat{L}\right)^{-n}, \tag{5}$$

from which follows, that

$$\frac{\partial \varrho(t)}{\partial t} = \hat{L}[\varrho(t)] \tag{6}$$

known as the master equation.

Let us consider an operator $\hat{L}$ acting on the Hilbert space of a qubit. The question arises, under what condition can this operator represent a generator of the dynamical semigroup. The answer was given by Lindblad [19], and by Gorini, Kossakowski and Sudarshan (GKS) [20]. We use the notation of the latter authors. According to this, the most general form of a generator of a Markovian semigroup reads

$$\hat{L}(\varrho) = -i[\hat{H}, \varrho] + \frac{1}{2}\sum_{i,j=1}^{3} C_{i,j}\left([\hat{\sigma}_i\varrho, \hat{\sigma}_j] + [\hat{\sigma}_i, \varrho\hat{\sigma}_j]\right), \tag{7}$$

where the $\hat{\sigma}$'s are the Pauli-matrices. The first contribution on the right hand size describes a possible unitary evolution, the Hamiltonian $\hat{H}$ being a Hermitian matrix which can be chosen to be traceless without the loss of generality. The second contribution describes the stochastic part of the evolution. The Hermitian positive semidefinite matrix $C_{i,j}$ is called the GKS matrix, and it contains all the information on the nature of the dynamics. Note that in order to preserve the trace of the density matrix throughout the generated evolution, $\hat{L}\varrho$ should be traceless. In fact, equation (7) describes the most general linear operator of this kind.

In the following we describe how to extract the Hamiltonian $\hat{H}$ and the GKS matrix C, if one is given an arbitrary function $\hat{L}\varrho$ of a single-qubit density matrix. In order to do so we utilize the relation between the GKS matrix and the affine representation mentioned also in [10].

A generic density matrix can be expanded on the basis of the three Pauli-matrices and the unit operator, obtaining its usual real 3-vector representation $\underline{r}$ displayable on the unit radius Bloch sphere:

$$\underline{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}, \quad r_i = \mathrm{tr}(\varrho\hat{\sigma}_i), \tag{8}$$

so that

$$\varrho = \frac{1}{2}\left(\hat{1} + \sum_{i=1}^{3} r_i\hat{\sigma}_i\right). \tag{9}$$

Direct calculation shows that the real 3-vector $\hat{L}[\underline{r}]$ corresponding to the most general $\hat{L}\varrho$ in equation (7), reads as

$$\begin{aligned}\hat{L}[\underline{r}] = &\begin{pmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{pmatrix}\underline{r} \\ &+ \begin{pmatrix} -2(C_{22}+C_{33}) & C_{12}+C_{21} & C_{13}+C_{31} \\ C_{12}+C_{21} & -2(C_{11}+C_{33}) & C_{23}+C_{32} \\ C_{13}+C_{31} & C_{23}+C_{32} & -2(C_{11}+C_{22}) \end{pmatrix}\underline{r} \\ &+ 2i\begin{pmatrix} C_{23}-C_{32} \\ C_{31}-C_{13} \\ C_{12}-C_{21} \end{pmatrix},\end{aligned} \tag{10}$$

where

$$\hat{H} = \frac{1}{2}\sum_{i=1}^{3} h_i\hat{\sigma}_i. \tag{11}$$

Thus $\hat{L}$ appears as an affine linear mapping in the 3 space, the above mapping is called the affine representation of $\hat{L}$.

If we are now given an arbitrary function $\hat{L}$ of ϱ, so that $\hat{L}\varrho$ is traceless and depends on the components of the arbitrary density matrix ϱ linearly, we can find the $\underline{r'}$ corresponding to $\hat{L}\varrho$ according to equation (8) as a function of the components of $\underline{r}$ representing ϱ. This is a linear inhomogeneous operator, which can be always decomposed into a sum of a homogeneous antisymmetric operator, a homogeneous symmetric operator and a vector representing the inhomogeneity. For qubits, this decomposition of the affine representation of the generator is quite meaningful: according to equation (10), the information on the unitary part of the generator, i.e. the Hamiltonian is encoded into the antisymmetric part of this operator, while the real and imaginary parts can be found from the symmetric part and the inhomogeneity respectively.

In the absence of the inhomogeneity the generator is zero for the identity operator: $\hat{L}\hat{I} = 0$. Therefore the evolution is unital: it it preserves the completely mixed state: $\exp(\hat{L}t)\hat{I} = \hat{I}$. The inhomogeneity appears in the complex part of the elements of the GKS matrix. Thus for qubits, real GKS matrices correspond to unital dynamics.

Thus equipped with equation (10), we have the recipe how to find the standard GKS form of a generator of a dynamical semigroup for a quantum bit.

3 Simulation of infinitesimal generators

We intend to simulate the infinitesimal generator $L\hat{\varrho}$ of Markovian dynamics, on a single quantum bit. This can be understood in several ways. As an elementary step we consider the application of a quantum processor: an arrangement of quantum logic gates, and possibly measurements, acting on the qubit in argument, and certain ancillary systems. The ancillary systems can be used as a "program register": their state can influence the action of the processor on the qubit in argument, which constitutes a "data bit" in this context.

The next question might be, how to interpret the time. A possible generic approach would be to regard the single run of the processor as a *finite* time step Δt. The repeated application of the processor on the data bit results then in a discrete time evolution. One can then examine if this is a stroboscopic version of some valid continuous time evolution, and search for the proper master equation, as it was done e.g. in [13,14]. Here we adopt a simpler interpretation: we expect a single run of processor to simulate an *infinitesimal* time step:

$$\varrho_{\mathrm{out}} = \varrho_{\mathrm{in}} + \hat{L}\varrho_{\mathrm{in}}dt + \mathcal{O}(dt^2), \tag{12}$$

where dt should be encoded in the $|\Psi_{\mathrm{prog}}\rangle$ of the program register. The entire evolution can then be approximated with some accuracy by running this process many many times. This implements the equidistant first order Euler method [21] of solving equation (6), but time step, and thereby the *accuracy* is encoded quantum mechanically. Of course, the simulation is completely accurate if $dt \to 0$, and the number of repetitions tends to infinity. We remark here, that simulation of decoherence mechanisms with an array of quantum gates has proven to be fruitful in other problems too [22,23].

Physically, the simulation scheme can be envisaged as a simple collision model: the data qubit is represented by a physical system localized in space. It interacts with flying program bits represented by e.g. spin of particles emerging from an oven. Each program bit causes the system to evolve a small time step further.

It follows from equation (12), that there must exist a program state, for which $dt = 0$, and thus $\varrho_{\mathrm{out}} = \varrho_{\mathrm{in}}$, in the quantum processor terminology we would say the processor implements the unit operator. It is a natural requirement for this kind of semigroup simulation. Thus our scheme is to some extent similar to the idea of simulating a reservoir with beam-splitters of transmittance around unity in quantum optics [24].

$$\mathcal{R}(\theta,\phi,\psi) = \begin{pmatrix} -\sin\phi\sin\psi + \cos\theta\cos\phi\cos\psi & -\cos\theta\cos\phi\sin\psi - \sin\phi\cos\psi & \sin\theta\cos\phi \\ \cos\theta\sin\phi\cos\psi + \cos\phi\sin\psi & \cos\phi\cos\psi - \cos\theta\sin\phi\sin\psi & \sin\theta\sin\phi \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{pmatrix} \in SO(3). \tag{21}$$

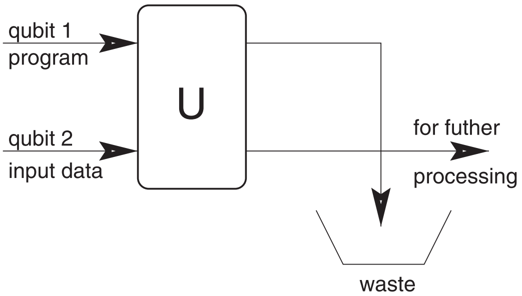

Fig. 1. The quantum network for the deterministic scheme. A single run of the network simulates an infinitesimal time-step in the evolution.

4 A scheme without a measurement

First we consider a simple deterministic scheme with a single ancillary qubit, as depicted in Figure 1. The first bit contains the program, which is most generally

$$|\Psi_{\text{prog}}\rangle = \sqrt{1-\varepsilon}e^{i\chi}|0\rangle + \sqrt{\varepsilon}|1\rangle, \quad 0 \le \varepsilon \le 1,\ \chi \text{ real.} \tag{13}$$

We don't consider mixed program states here, we want to study how decoherence associated with Markovian processes appears purely quantum-mechanically, and a mixed program state would imply some prescribed classical stochasticity. The input data is in bit 2, in the state ϱ_{in}. After the operation of the processor, the contents of the program register are dropped, and bit 2 contains the output

$$\varrho_{\text{out}} = \text{Tr}_1\left(\hat{U}(|\Psi_{\text{prog}}\rangle\langle\Psi_{\text{prog}}| \otimes \varrho_{\text{in}})U^\dagger\right) \tag{14}$$

which is passed for further processing.

To have the identity operator implemented, there should be a program state for which $\varrho_{\text{out}} = \varrho_{\text{in}}$ holds. We chose this to be the program state $|0\rangle$. The most general processor that is possible under such circumstances is a controlled U gate:

$$\hat{U} = \begin{pmatrix} \hat{1} & 0 \\ 0 & \hat{U}_2 \end{pmatrix}. \tag{15}$$

This does nothing to the second qubit if the first (control) one is in the state $|0\rangle$, while it carries out the $SU(2)$ operation

$$\hat{U}_2 = \begin{pmatrix} \cos(\frac{\theta}{2})e^{-i\frac{\phi+\psi}{2}} & -\sin(\frac{\theta}{2})e^{-i\frac{\phi-\psi}{2}} \\ \sin(\frac{\theta}{2})e^{i\frac{\phi-\psi}{2}} & \cos(\frac{\theta}{2})e^{i\frac{\phi+\psi}{2}} \end{pmatrix} \tag{16}$$

in the lower right block of its matrix, if the first qubit is in state $|1\rangle$. We use the standard Euler angle parametrization in the y convention [25].

Now we can evaluate equation (12) with a generic input data state of equation (9), and program of equation (13), to obtain the effect of a single run of the processor. Note that due to the orthogonality of the program states corresponding to $\hat{1}$ and $\hat{U}_2$ (which is a consequence of the unitarity of $\hat{U}$), equation (14) simplifies to

$$\varrho_{\text{out}} = (1-\varepsilon)\varrho_{\text{in}} + \varepsilon\hat{U}_2\varrho_{\text{in}}\hat{U}_2^\dagger. \tag{17}$$

Thus from the point of view of the effect on the input state, the quantum circuit does nothing else than apply the unitary operation $\hat{U}_2$ on the input state, with the probability ε. The difference is that if the operation is realized by the quantum circuit, then the information which disappears from the system will be stored in the dropped quantum state of the program register. In the other case the information will be entirely classical, one bit per infinitesimal timestep: we can be aware, whether the operation $\hat{U}_2$ was carried out or not. Note that in both of the cases the process is reversible, provided that we have the appropriate information at hand.

According to equation (17), we can write

$$\varrho_{\text{out}} = \varrho_{\text{in}} + (\hat{U}_2\varrho_{\text{in}}\hat{U}_2^\dagger - \varrho_{\text{in}})\varepsilon. \tag{18}$$

Thus we identify $\varepsilon = dt$, and comparing with equation (12) we get

$$\hat{L}\varrho_{\text{in}} = \hat{U}_2\varrho_{\text{in}}\hat{U}_2^\dagger - \varrho_{\text{in}}. \tag{19}$$

The transformation on the Bloch ball corresponding to $\hat{U}_2\varrho\hat{U}_2^\dagger$ is a rotation of the vector $\underline{r}$ corresponding to ϱ

$$\underline{r}' = \mathcal{R}(\theta,\phi,\psi)\underline{r}, \tag{20}$$

where $\mathcal{R}$ is the appropriate element of the adjoint representation of $SU(2)$:

see equation (21) *above.*

For the generator in equation (19) we have thus

$$\hat{L}[\underline{r}] = \left(\mathcal{R}(\theta,\phi,\psi) - \hat{1}\right)\underline{r}. \tag{22}$$

This can be compared with equation (10) to extract the properties of the generator. The transformation in equation (20) is homogeneous, thus the generated dynamics is unital. In addition, the generator is zero for the eigenstates of $\hat{U}_2$, therefore the line in the Bloch sphere formed by the mixture of these eigenstates is preserved by the evolution.

As for the coherent part of the evolution, we get the Hamiltonian

$$\hat{H} = \begin{pmatrix} \sin(\phi+\psi)\cos^2\frac{\theta}{2} & -\frac{i}{2}\sin\theta\left(e^{-i\phi}+e^{i\psi}\right) \\ \frac{i}{2}\sin\theta\left(e^{i\phi}+e^{-i\psi}\right) & -\sin(\phi+\psi)\cos^2\frac{\theta}{2} \end{pmatrix}. \tag{23}$$

This is zero if $\phi+\psi = (2k+1)\pi$ or $\theta = \pi + 2k\pi$ holds, which is equivalent to $\text{tr}U_2 = 0$. Thus in case of traceless

$$C_{\theta,\phi,\psi} = \begin{pmatrix} \sin^2\frac{\theta}{2}\sin^2\frac{\phi-\psi}{2} & \frac{1}{2}\sin^2\frac{\theta}{2}\sin(\psi-\phi) & \frac{1}{4}\sin\theta(\cos\phi-\cos\psi) \\ \frac{1}{2}\sin^2\frac{\theta}{2}\sin(\psi-\phi) & \sin^2\frac{\theta}{2}\cos^2\frac{\phi-\psi}{2} & \frac{1}{4}\sin\theta(\sin\phi+\sin\psi) \\ \frac{1}{4}\sin\theta(\cos\phi-\cos\psi) & \frac{1}{4}\sin\theta(\sin\phi+\sin\psi) & \cos^2\frac{\theta}{2}\sin^2\frac{\phi+\psi}{2} \end{pmatrix} \quad (24)$$

unitaries, we obtain a purely stochastic evolution in the sense that it lacks the Hamiltonian part. Evaluating the symmetric part of equation (21), and comparing with (10), we obtain the GKS matrix:

see equation (24) *above.*

Specifically, if $\hat{U}_2$ is traceless because $\psi = \pi - \phi$ holds, we obtain from equation (24) the following GKS matrix:

$$C_{\theta,\phi} = \begin{pmatrix} \sin^2\frac{\theta}{2}\cos^2\phi & \frac{1}{2}\sin^2\frac{\theta}{2}\sin(2\phi) & \frac{1}{2}\sin\theta\cos\phi \\ \frac{1}{2}\sin^2\frac{\theta}{2}\sin 2\phi & \sin^2\frac{\theta}{2}\sin^2\phi & \frac{1}{2}\sin\theta\sin\phi \\ \frac{1}{2}\sin\theta\cos\phi & \frac{1}{2}\sin\theta\sin\phi & \cos^2\frac{\theta}{2} \end{pmatrix}. \quad (25)$$

The matrix in equation (25) can be interpreted as follows. Consider a unitary operator $\mathcal{U} \in SU(2)$ acting on the qubit's Hilbert space, and a superoperator $\mathcal{E}$, which describes unitary evolution described by the GKS matrix C. According to reference [10] *unitary conjugation* of a superoperator $\mathcal{E}$, that is,

$$\mathcal{E}' = \mathcal{U}^\dagger \mathcal{E} \mathcal{U}, \quad (26)$$

where $\mathcal{U}(\varrho) = \mathcal{U}\varrho\mathcal{U}^\dagger$ yields another superoperator describing Markovian dynamics as well. The resulting GKS matrix is

$$C' = \mathcal{R} C \mathcal{R}^T, \quad (27)$$

where $\mathcal{R}$ is the element of $SO(3)$, the adjoint representation of $SU(2)$ corresponding to $\mathcal{U}$, and the T stands for transposition. Thus $\mathcal{R}$ is a real 3-rotation, which can be visualized as a rotation in the Bloch-sphere picture. The effect of unitary conjugation is to apply the same operation on a transformed basis. In the actual case, the matrix in equation (25) can be rewritten as

$$C_{\theta,\phi} = \mathcal{R}(\theta/2,\phi,\psi) \begin{pmatrix} 0 & & \\ & 0 & \\ & & 1 \end{pmatrix} \mathcal{R}^T(\theta/2,\phi,\psi). \quad (28)$$

The matrix between the two rotations defines a phase damping about the z-axis. Thus the process is a phase damping about an axis pointing towards the spherical polar angles $\theta/2, \phi$. In fact the rotation U_2 moves the z-axis to the direction described by the spherical polar angles θ, ϕ, so the direction of the phase damping is "half way" between the z axis, and its transform by U_2. Note that the angle ψ is irrelevant in equation (28), as it does not influence the polar angle of the rotated z-axis. The phase damping channel has a single direction which is special, thus it is necessarily described by two parameters. If $\hat{U}_2$ is traceless because $\theta = \pi + 2k\pi$ is satisfied, we obtain a phase damping channel about an axis in the xy-plane. We can conclude that the controlled U gate with a traceless U_2 is capable of simulating a generator of an arbitrary phase damping channel in our scheme. This is consistent with the fact that the dynamics should preserve the line representing the mixtures of the eigenstates of U_2: the above obtained directions of the axis of the phase damping channel indeed point towards this direction.

Returning to the generic GKS matrix in equation (24) we find that rank $C_{\theta,\phi,\psi} = 1$, thus this general GKS matrix also describes a phase damping channel, physically the same process as in the traceless case. The only difference is that the evolution is now accompanied by a coherent part, generated by the Hamiltonian in equation (23).

It is interesting to compare the discussed scenario with the stroboscopic approach of the other collision-like models of decoherence [13,14]. In that case a single run of a processor realizes a *finite* time step Δt, thus after the nth run of the processor the output is the density matrix at $t_n = n\Delta t$. Then the so defined t_n is replaced by a continuous parameter, obtaining the target evolution. The simulated evolution will exactly coincide the target evolution at time steps t_n. It turns out that the controlled U gate simulates a phase damping channel in that case, too. The question to be answered in order to compare with the two scenarios is the following one: assuming the above interpretation of the operation of our processor, and setting $dt = \Delta t \to 0$, will we obtain the same evolution from the two different considerations?

To examine this, we chose the basis formed by the eigenstates of $\hat{U}_2$ in the data Hilbert space, and use the notation

$$\hat{U}_2 = \begin{pmatrix} e^{i\alpha_1} & 0 \\ 0 & e^{i\alpha_2} \end{pmatrix}. \quad (29)$$

We do not change the basis on the control (program) qubit's space. Using the program state in equation (13), the operation of a single run on the processor according to equation (17) reads

$$\varrho_{\text{out}} = \begin{pmatrix} \varrho_{\text{in},00} & \varrho_{\text{in},01}(1+\varepsilon(e^{i\alpha}-1)) \\ \varrho_{\text{in},10}(1+\varepsilon(e^{-i\alpha}-1)) & \varrho_{\text{in},11} \end{pmatrix}, \quad (30)$$

where $\alpha = \alpha_1 - \alpha_2$. From this it follows that after the n-fold application, setting $n = t/\Delta t$ we have

$$\begin{pmatrix} \varrho_{\text{in},00} & \varrho_{\text{in},01}(1+\varepsilon(e^{i\alpha}-1))^{\frac{t}{\Delta t}} \\ \varrho_{\text{in},10}(1+\varepsilon(e^{-i\alpha}-1))^{\frac{t}{\Delta t}} & \varrho_{\text{in},11} \end{pmatrix}. \quad (31)$$

Following the considerations in reference [14], the generator of the above evolution has the GKS matrix with a single nonzero element $C_{33} = -\frac{1}{2}\ln|1+\varepsilon(e^{i\alpha}-1)|/\Delta t$, and an effective Hamiltonian with $h_3 = \arg(1+\varepsilon(e^{i\alpha}-1))/\Delta t$.

It appears that similarly to the case of our previous considerations, $h_3 = 0$ if $\alpha = \pi + 2k\pi$, thus the coherent part of the evolution disappears again if U_2 is traceless

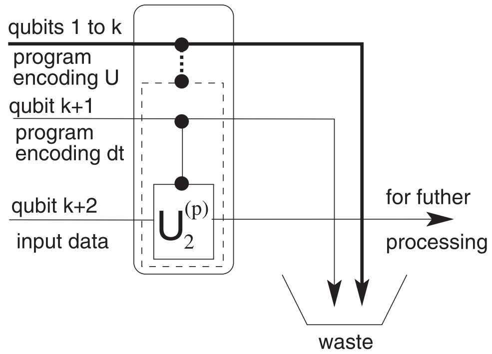

Fig. 2. A Nielsen-Chuang type quantum processor which implements different controlled-U gates. The last qubit of the program register (qubit $k+1$) is reserved for encoding the length of the time step as described in the case of a single controlled-U gate in Section 4. By preparing the first k (program) qubits in a state of a given orthonormal basis one can chose between $p = 1 \ldots 2^k$ different $U_2^{(p)}$ characterizing the chosen controlled-U gate. Thereby the properties of the simulated Liouvillian superoperator can be controlled by the program state.

(except for the trivial case $\alpha = 2k\pi$, which results in no change of the state).

As for the stochastic part of the evolution, consider now the case of traceless U_2, e.g. $\alpha = \pi$. It is easy to calculate the limit of C_{33}:

$$-\lim_{\Delta t\to 0} \frac{1}{2}\frac{\ln|1-2\varepsilon|}{\Delta t}\bigg|_{\varepsilon=\Delta t} = 1. \tag{32}$$

This is to be compared with the GKS matrix element in equation (28). Now we are working in the basis of the eigenvectors of $\hat{U}_2$, thus we expect the GKS matrix to contain a single nonzero value, $C_{33} = 1$. This is indeed the case in the limit when each run represents an infinitesimal time step $\Delta t \to 0$ of length $\Delta t = \varepsilon$, the parameter of the program state. Thus we obtain the same evolution on the same time scale: a phase damping towards the eigenstates of $\hat{U}_2$ as the one we had from the previous, simpler considerations. While in the case of the stroboscopic approach the inaccuracy of the simulation arises from the absence of the appropriate state between the discrete time instants simulated, in our case the error appears in each time step, depending on the size of $dt = \varepsilon$ chosen. The comparison shows that the simulation of the infinitesimal generator as described in this section in case of $dt = \varepsilon \to 0$ indeed produces the desired Markovian evolution, and it is consistent with the other collision models.

One can increase the number of the parameters of the evolution encoded into a program state by utilizing a Nielsen-Chuang type processor which is capable of implementing a larger (though finite) number of unitaries. E.g. the processor circuit used in [7] implements the three Pauli-operations in addition to the unity operation. By choosing the appropriate program state, one can chose between different controlled U gates in each time step. It is possible even to obtain the same to chose between different *controlled U gates*, as depicted in Figure 2. In this case the first k qubits serve the purpose of choosing from 2^k different controlled-U gates, while the $(k+1)$th qubit (the last one of the program register) plays exactly the role of the program qubit encoding the time step, as described in before in this Section. This introduces the possibility of implementing a linear (or Trotter) combination [10] of different phase damping channels.

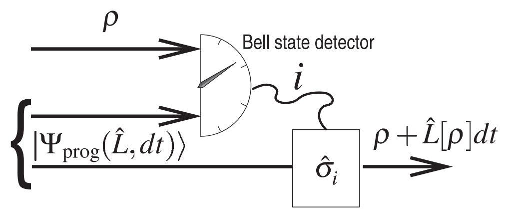

Fig. 3. Bennett's teleportation scheme as a programmable circuit for Markovian decoherence

5 Control via teleportation

In this section we briefly describe another simulation scheme based on Bennett's quantum teleportation. It is depicted in Figure 3. The initial state ϱ impinges at the input of a teleportation arrangement. Two additional qubits are prepared in an entangled state

$$|\Psi_{\text{prog}}\rangle = \sqrt{1-\varepsilon}|B_0\rangle + \sqrt{\varepsilon}\left(\alpha_1|B_1\rangle + \alpha_2|B_2\rangle + \alpha_3|B_3\rangle\right), \tag{33}$$

with $|\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$, which serves as the *program state*. We use the notation

$$\begin{aligned}
|B_0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle|1\rangle - |1\rangle|0\rangle);\\
|B_1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle|0\rangle - |1\rangle|1\rangle);\\
|B_2\rangle &= \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle);\\
|B_3\rangle &= \frac{1}{\sqrt{2}}(|0\rangle|1\rangle + |1\rangle|0\rangle).
\end{aligned} \tag{34}$$

for the elements of the Bell basis. Then the usual Bennett teleportation is carried out: a Bell state measurement on the two appropriate qubits is carried out, and depending on the result, the appropriate unitary transformation $\hat{\sigma}_i$ (the identity operator or one of the Pauli operators) is carried out.

For $\varepsilon = 0$ we have $|\Psi_{\text{prog}}\rangle = |B_0\rangle$, the state ϱ is simply teleported: the identity operator is implemented. However, for nonzero ε we obtain

$$\varrho' = (1-\varepsilon)\varrho + \varepsilon\left(|\alpha_1|^2\hat{\sigma}_1\varrho\hat{\sigma}_1 + |\alpha_2|^2\hat{\sigma}_2\varrho\hat{\sigma}_2 + |\alpha_3|^2\hat{\sigma}_3\varrho\hat{\sigma}_3\right) \tag{35}$$

as a "teleported" state. Thus the scheme is essentially equivalent to a random application of the Pauli operators. Setting $dt = \varepsilon$ as in the previous section, we can define

$$\hat{L}\varrho = |\alpha_1|^2\hat{\sigma}_1\varrho\sigma_1 + |\alpha_2|^2\hat{\sigma}_2\varrho\sigma_2 + |\alpha_3|^2\hat{\sigma}_3\varrho\sigma_3 - \varrho. \tag{36}$$

A straightforward calculation shows that the corresponding GKS matrix reads

$$\mathcal{C}(\alpha_1,\alpha_2,\alpha_3)=\begin{pmatrix}|\alpha_1|^2 & & \\ & |\alpha_2|^2 & \\ & & |\alpha_3|^2\end{pmatrix}. \qquad (37)$$

We have obtained a GKS matrix of rank 3, describing a generic Pauli channel. In this case we have the parameters of the dynamics encoded into the program state, too.

It is worth noting here that this second scheme is indeed irreversible. However, the same process could be simulated in a similar reversible framework as in Section 4, utilizing two ancillary qubits, and the universal programmable quantum gate array in references [6,7].

6 Summary

We have investigated quantum computational schemes for the simulation of infinitesimal time steps, and in this way, the generator of Markovian dynamics on a qubit, where the value of the infinitesimal time step is encoded in a quantum state at the input of the device.

We have found that using a controlled U gate as the programmable quantum circuit, a phase damping about an arbitrary axis can be simulated. We shown that this model is interpolable with a different interpretation of a collisional model of decoherence utilizing the same "quantum hardware". Our scheme relies on the realization of the transformation in equation (17) of the input quantum state by the controlled U gate, which is the simplest quantum processor of the Nielsen-Chuang type. A possible extension is to consider a more general quantum processor of this kind with a larger program space. This is capable of implementing a discrete set of distinct unitary operations on the data quantum bit in a way that each implemented unitary operator is assigned a state of an orthogonal basis in the program space. Thus in addition to the size of the time step, a the choice of the unitary equation (17) from a given set is encoded into the program state, too. This can be even done in a way that one has a qubit exactly reserved for the encoding of the size of the timestep, as depicted in Figure 2. Thereby the possibility of encoding the parameters of the evolution (that is, the Liouvillian superoperator) into the program state prevails. Furthermore, by changing the program state in each step of the evolution properly, a linear (Trotter) combination of Markovian dynamics can be simulated, too.

We have also considered a Bennett quantum teleportation scheme, which can be implemented as a programmable quantum circuit, too, as a possible quantum circuit for performing a similar task. We have found the capabilities of this measurement-based closed loop scheme similar to that of the deterministic quantum processors.

The described scenario, as the other similar microscopic models, provides a framework offering many possibilities of generalization ranging from systems of larger dimensionality to the inclusion of a quantum memory and thereby accessing non-Markovian processes, too. Unlike in the case of a classical simulation of quantum dynamics, the present scheme works for any, even unknown initial state of the system qubit, which may emerge as an output of another quantum computation. We believe that the study of simple quantum systems such as those described here facilitates the understanding decoherence in general.

This work was supported by the European Union projects QGATES and CONQUEST, by the Slovak Academy of Sciences via the project CE-PI, by the project APVT-99-012304, and by the Research Fund of Hungary (OTKA) under contracts Nos. T043287 and T034484. The authors acknowledge the relevant contribution of Dr. Ákos László at Institute of Mathematics, University of Pécs to this work, who died unexpectedly at his age of 32.

References

1. Yu. I. Manin, *Vychislimoe i nevychislimoe* (Sovetskoye Radio, Moscow, 1980)
2. R.P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982)
3. K. Krauss, *States, Efffects and Operations: Fundamental Notions of Quantum Theory* (Springer Verlag, Berlin, 1983)
4. W.F. Stinespring, Proc. Am. Math. Soc. **6**, 211 (1955)
5. M.A. Nielsen, I.L. Chuang, Phys. Rev. Lett. **79**, 321 (1997)
6. M. Hillery, V. Bužek, M. Ziman, Fortsch. Phys. Prog. Phys. **49**, 987 (2001)
7. M. Hillery, V. Bužek, M. Ziman, Phys. Rev. A **65**, 022301 (2002)
8. G. Vidal, L. Masanes, J.I. Cirac, Phys. Rev. Lett. **88**, 047905 (2002)
9. M. Hillery, M. Ziman, V. Bužek, Phys. Rev. A **66**, 042302 (2002)
10. D. Bacon et al., Phys. Rev. A **64**, 062302 (2001)
11. V. Scarani, M. Ziman, P. Štelmachovič, N. Gisin, V. Bužek, Phys. Rev. Lett. **88**, 097905 (2002)
12. M. Ziman, P. Štelmachovič, V. Bužek, M. Hillery, V. Scarani, N. Gisin, Phys. Rev. A **65**, 042105 (2002)
13. M. Ziman, P. Štelmachovič, V. Bužek, Open Syst. Inf. Dyn. **12**, 81 (2005)
14. M. Ziman, V. Bužek, e-print `arXiv:quant-ph/0505040` (2005)
15. L. Diósi, T. Feldmann, R. Kosloff, e-print `arXiv:quant-ph/0505219` (2005)
16. Ch. Bennett et al., Phys. Rev. Lett. **70**, 1895 (1993)
17. G. Bowen, S. Bose, Phys. Rev. Lett. **87**, 267901 (2001)
18. R. Alicki, K. Lendi, *Quantum Dynamical Semigroups and Applications*, Lecture notes in Physics (Springer-Verlag, Berlin, 1987)
19. G. Lindblad, Comm. Math. Phys. **48**, 119 (1976)
20. V. Gorini, A. Kossakowski, E.C.G. Sudarshan, J. Math. Phys. **17**, 821 (1976)
21. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical recipes in Fortran 77* (Cambridge University press, Cambridge, 1992)
22. V. Scarani et al., Phys. Rev. Lett. **88**, 097905 (2002)
23. M. Ziman et al., Phys. Rev. A **65**, 042105 (2002)
24. M.S. Kim, N. Imoto, Phys. Rev. A **52**, 2401 (1995)
25. E.W. Weisstein: "*Euler Angles*" From MathWorld–A Wolfram Web Resource `http://mathworld.wolfram.com/EulerAngles.html`