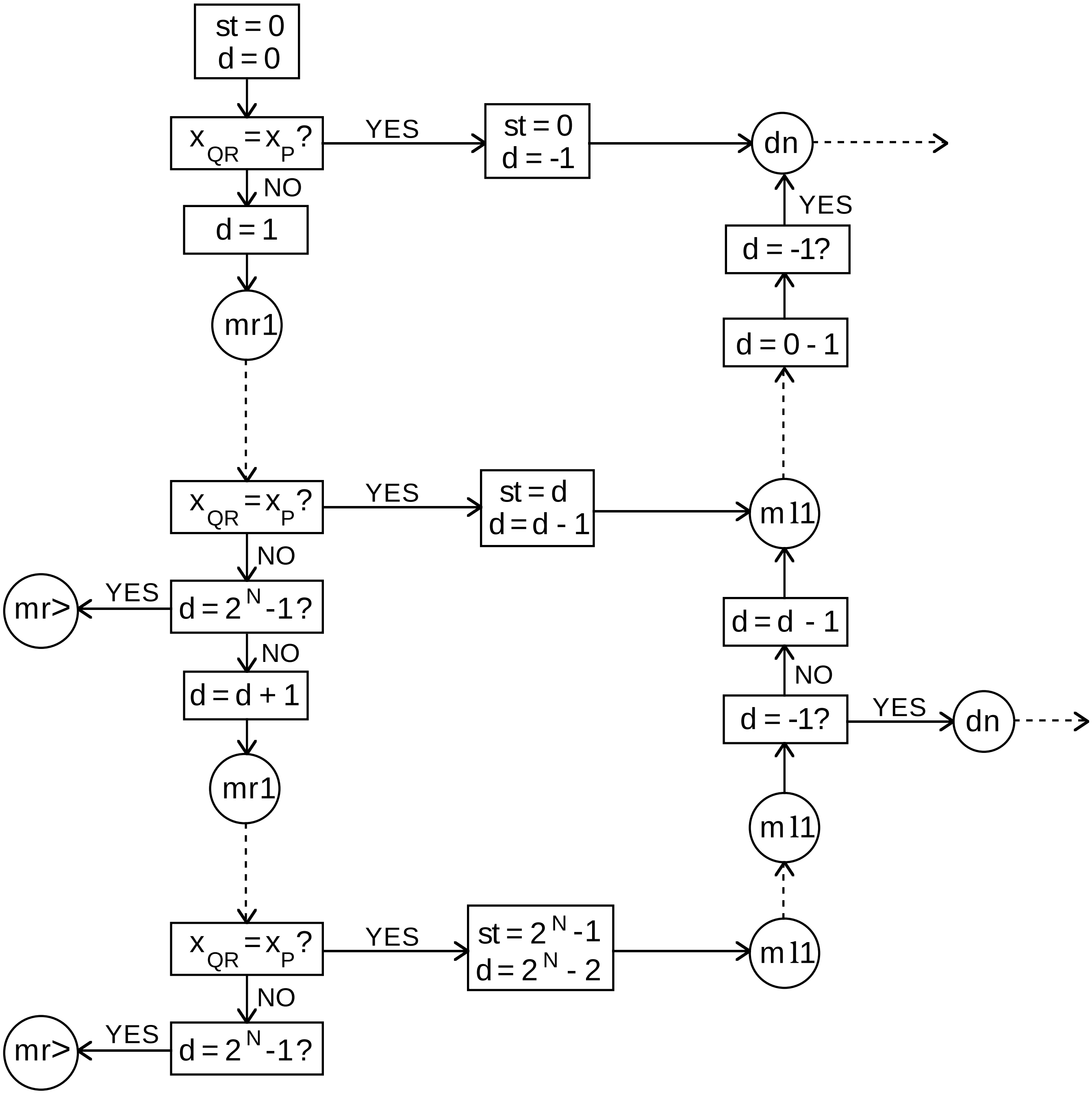
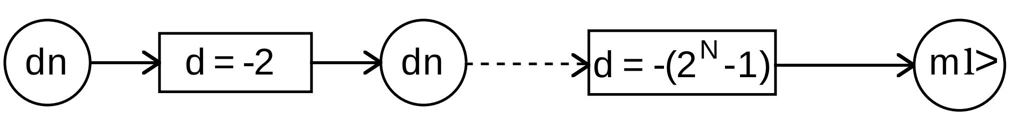


Ballast Part



Quantum Robots Plus Environments

Paul Benioff
Physics Division, Argonne National Laboratory
Argonne, IL 60439
e-mail: pbenioff@anl.gov
(April 4, 2007)

A quantum robot is a mobile quantum system, including an on board quantum computer and needed ancillary systems, that interacts with an environment of quantum systems. Quantum robots carry out tasks whose goals include making specified changes in the state of the environment or carrying out measurements on the environment. The environments considered so far, oracles, data bases, and quantum registers, are seen to be special cases of environments considered here. It is also seen that a quantum robot should include a quantum computer and cannot be simply a multistate head.

A model of quantum robots and their interactions is discussed in which each task, as a sequence of alternating computation and action phases,is described by a unitary single time step operator $\mathcal{T} = \mathcal{T}_a + \mathcal{T}_c$ (discrete space and time are assumed). The overall system dynamics is described as a sum over paths of completed computation ($\mathcal{T}_c$) and action ($\mathcal{T}_a$) phases. A simple example of a task, measuring the distance between the quantum robot and a particle on a 1D lattice with quantum phase path dispersion present, is analyzed. A decision diagram for the task is presented and analyzed.

I. INTRODUCTION

Quantum computers are of much interest due to their increased power over classical computers in solving certain problems [1,2]. Most studies of quantum computers consider them as stand alone systems operating in isolation from external systems as an environment. So far work on quantum computers which includes interactions with environments is limited mainly to noise effects, data base searching, and quantum oracle computing. The former considers environmental interactions as a source of noise and errors. This has stimulated the development of quantum error correcting codes to minimize this effect [3,4] Other methods rely on the use of properties of systems with relatively long decoherence times [5,6]. Quantum oracle computing has been discussed but not developed extensively [7]. Data base searching has been much discussed recently [2].

Here the emphasis is on quantum computers and their interactions with the environment in general. The interest is in quantum computers along with ancillary systems that can move in and interact with an environment of quantum systems. These are the defining characteristics of quantum robots.

Quantum robots are of interest from a foundational viewpoint [8]. If quantum mechanics is universally valid, then the systems that carry out theoretical calculations (computers) and physical experiments to test theoretical predictions (robots) must be described within quantum mechanics, i. e. as quantum computers and quantum robots. It follows that the systems that test the validity of quantum mechanics must be described by the same theory they are testing. Quantum mechanics must describe its own validity to the maximum extent possible [9].

A related reason that supports study of quantum robots and their interactions with an environment is that they provide a *very small* first step towards a quantum mechanical description of systems that are aware of their environment, make decisions, are intelligent, and create theories such as quantum mechanics [10–12]. If quantum mechanics is universal, then these systems must be described with quantum mechanics to the maximum extent possible.

Another reason that supports study of quantum robots is that there is no limitation on the types of environments included. Environments studied so far, such as oracles, data bases, and quantum registers, are special types of environments. These specific types of environments are discussed in the next section. Reasons are also given why the quantum robot must include a quantum computer and cannot be simply a multistate head.

Section III summarizes a dynamical model of the interactions of quantum robots with environments that has been described elsewhere [8]. The dynamics are described in terms of tasks carried out by the quantum robot. Tasks are defined as sequences of alternating computation and action phases. The model describes task dynamics in terms of iteration of step operators and Feynman [13] sums over phase paths.

A simple example of a task, measuring the distance between a quantum robot and a particle, is described in Section IV. The example is a generalization of the one described elsewhere in that sums over different paths of states are included in the phase path sum. A description of the task including the steps needed is given along with a representation of the task as a decision diagram. Accuracy conditions are also discussed. The complexity of even a simple measurement, as a task for a quantum robot, and the relation to a possible Church Turing hypothesis [15,16] applicable to the carrying out of physical experiments [8] is noted in the last section.

II. ORACLES, DATA BASES, QUANTUM REGISTERS,AND HEADS

Oracles and data bases, as used in quantum oracle computation [7] and in Grover's algorithm [2], and quantum registers, are special cases of environments considered here. Oracles are special cases because the relevant properties are not supposed to be time dependent. An oracle that answered yes to a question at one time and no to the same question later on would be regarded as defective. Also the answer to a question Q at time t should not be influenced by the asking of another question P at an earlier time t_1 where both the yes and no answers to P are ignored.

As environmental systems the states of both data bases and quantum registers can be time dependent in the sense that data bases change as old data is replaced or corrected or new data is added. Also the states of quantum registers change with time as part of any quantum computation process. However in both these cases the systems change in carefully specified ways. In particular, neither of these systems are supposed to change spontaneously in the absence of interaction with external systems. An environment of moving interacting systems whose states changed as a result of the interactions or motion would not serve as a physical model of data bases or quantum registers. In the latter case the requirement of no spontaneous changes is emphasized by the need for quantum error correction codes and other stabilization methods to minimize this effect [3,4].

Environments considered here are not so limited. They include interacting moving systems whose quantum states, $\Psi(t) = e^{-iHt}\Psi(0)$, are evolving with time. For these systems the yes answer to any question, represented by a projection operator Q, has a time dependent probability $\langle\Psi(t)|Q|\Psi(t)\rangle$. Also let P and Q be two projection operators corresponding to question P asked at time t_1 and Q asked at a later time t. The probability of a yes answer to Q at t is not in general equal to the probability of a yes answer to Q at time t given that P was asked at time t_1 and the answer ignored. The latter probability is given by $Tr\rho(t)Q = Tr\rho(t_1)Q(t-t_1)$ where $Q(t-t_1) = e^{iH(t-t_1)}Qe^{-iH(t-t_1)}$ and $\rho(t_1) = PR_{\Psi(t_1)}P + (1-P)R_{\Psi(t_1)}(1-P)$ with $R_{\Psi(t_1)} = |\Psi(t_1)\rangle\langle\Psi(t_1)|$. This nonequality holds in the case that $Q(t-t_1)$ does not commute with P.

Arguments that the quantum robot must include a quantum computer and cannot be simply a multistate head are based on the number of degrees of freedom in the head. If the head is a single degree of freedom that must be responsive to at least N different alternatives of environmental information, it must be possible to distinguish between N different internal states of the head. For large values of N, as would be the case for a general purpose or universal quantum robot able to carry out many tasks in many different environments [8], this is physically unreasonable. For example if the head is a single spin system, it is very difficult to distinguish N different spin projection states.

In this case, and in any case where the number of alternatives that must be distinguished by the head is exponentially large, (e.g. distinguishing all bit strings of length n) the only reasonable approach is to allow the number of degrees of freedom in the head to be polynomial in $\log N$. But this is equivalent to requiring that the head include a quantum computer. That is it must be a quantum robot. The same argument holds if the head has a small number (> 1) of degrees of freedom.

III. A MODEL OF QUANTUM ROBOTS WITH ENVIRONMENTS

Here the description of a specific model of quantum robots with environments will be summarized. Some details are given elsewhere [8]. A quantum robot consists of an on board quantum computer, a finite state output system o and a control qubit c. The dynamics of thse systems and their environmental interactions can be described by tasks consisting of alternating computation and action phases. The goal of each computation phase is to determine the following action by generating a new state of o. Input consists of the former state of o, any stored memory, and observations of the neighborhood state of the environment. During the following action phase the action, determined by the state of o, is carried out. The states of all on board systems remain unchanged. Actions include motion of the quantum robot and neighborhood changes of the environment state. The function of the control qubit c is to turn on and off the two types of phases. The computation [action] phase is inactive when c is in state $|1\rangle[|0\rangle]$. Each phase terminates by changing the state of c.

A unitary step operator $T = T_a + T_c$ which describes the dynamical changes of the overall system during one time step is associated with each task. (To keep things simple discrete space and time are assumed.) If $\Psi(0)$ is the overall quantum robot plus environment state at time 0 the state after n time steps is given by $\Psi(n) = T^n\Psi(0)$. The action and computation phase step operators satisfy $T_a = TP_1^c$ and $T_c = TP_0^c$ where the projection operators P_i^c refer to the states of c.

Here T_c is such that it may depend on but not change the quantum robot location $\underline{x} = x, y, z$. This can be expressed by a diagonality condition

$$T_c = \sum_{\underline{x}} P^{qr}_{\underline{x}} T_c P^{qr}_{\underline{x}} P^c_0. \tag{1}$$

The requirement that T_a depend on but not change the states of o is given by a similar expression:

$$T_a = \sum_l P^o_l T_a P^o_l P^c_1. \tag{2}$$

The independence of T_a from the quantum computer states $|b\rangle$ is given by the commutativity of T_a with projection operators P^{qc}_b.

Both T_c and T_a make local changes only in the environment state. However those made by T_c are limited to entanglements with on board system states and other changes that are a direct result of the observation interaction. Changes made by T_a do not give such entanglements and are not limited to observation interactions. Details on locality requirements given in terms of neighborhoods of the quantum robot are given elsewhere [8].

The description of T given so far applies to motionless noninteracting environment systems for which the environment Hamiltonian $H_E = 0$. Extension to moving interacting systems can be done by replacing T by another step operator $\mathcal{T} = T_E^{1/2} T T_E^{1/2} = T_E^{1/2} T_a T_E^{1/2} + T_E^{1/2} T_c T_E^{1/2} = \mathcal{T}_a + \mathcal{T}_c$. Here T_a and T_c are as defined above and $T_E = e^{-iH_E\Delta}$ is a unitary step operator for the environment. This replacement of T by $\mathcal{T}$ becomes exact in the limit $\Delta \rightarrow 0$ [14].

It is useful to express the dynamical development of the system as a Feynman [13] sum over paths of computation and action phases, i.e. a phase path sum. To this end consider the matrix element $\langle w, i|\mathcal{T}^n|w_1, 0\rangle$ which gives the transition amplitude for going from state $|w_1, 0\rangle$ to state $|w, i\rangle$ in n steps. Here $|w\rangle$ denotes the state of all systems except that of the control qubit. One can use $\mathcal{T}^n = (\mathcal{T}(P^c_0 + P^c_1))^n$ to obtain

$$\langle w, i|\mathcal{T}^n|w_1, 0\rangle = \sum_{t=1}^{n} \sum_{\substack{paths\ p\ \ of \\ length\ t+1}} \sum_{h_1,\cdots,h_t=1}^{\delta(\sum,n)} \langle p(t+1), i|(\mathcal{T}_{v_t})^{h_t}|p(t)\rangle, \cdots, \langle p(3)|(\mathcal{T}_a)^{h_2}|p(2)\rangle\langle p(2)|(\mathcal{T}_c)^{h_1}|p(1), 0\rangle \tag{3}$$

Each term in this large sum give the amplitude for finding t alternating phases in the first n steps where the jth phase begins with all systems (except for c) in state $|p(j)\rangle$ and ends after h_j steps with all systems in state $|p(j+1)\rangle$. The upper limit on the h sums shows the restriction that the sum $h_1+, \cdots, +h_t = n$. The initial and final path states $|p(1)\rangle$ and $|p(t+1)\rangle$ are $|w_1\rangle$ and $|w\rangle$.

Eq. 3 is shown for the case that the initial phase is a computation phase, or c is in the initial state $|0\rangle$. A similar equation holds in case the initial phase is an action phase. The alternation of phases is expressed for this case by the subscripts v_j: if j is even $v_j = c$, if j is odd $v_j = a$. Also the restrictions of Eqs. 2 and 1 on T_a and T_c, which also hold for $\mathcal{T}_a$ and $\mathcal{T}_c$, show that if j is even, $|p(j)\rangle$ and $|p(j+1)\rangle$ show the same robot position states. If j is odd, $|p(j)\rangle$ and $|p(j+1)\rangle$ show the same quantum computer and o system states.

The equation shows clearly that for any n the overall system state is a linear sum over many phase path states of alternating computation and action phases for the task represented by $\mathcal{T}$. For each value of t and p, the equation gives the amplitude for the phase path p containing $t-1$ completed phases and one, the tth, which may or may not be complete. The h sums give the dispersion in the duration or number of time steps in each phase in p.

It follows from Eq. 3 that the overall system state $\Psi(n)$ can be expressed, for each initial state component, as an exponentially growing (with n) tree of phase paths. Each node in the tree corresponds to a state $|p(j)\rangle$. The t sum shows that some branches of the tree have very few nodes and include those having one node ($t = 1$).

IV. A SIMPLE EXAMPLE

A. Task Description

A simple example to illustrate the actions of a quantum robot has an environment with a single motionless particle p on a 1D lattice (i.e. $T_E = 1$ and $\mathcal{T} = T$). The task is to measure the distance between the quantum robot and p by alternating quantum robot motion with local observations for p and counting the number of nonobservations of p until p is found, (the search part). In the return part, the quantum robot goes back the same number of steps and the task ends by entering the ballast part. This part is present to preserve the unitarity of T.

The overall goal of the task, as a condition on T can be expressed as follows: Let $\phi = \sum_y c_y|y\rangle$ denote the state of p on the lattice with the quantum robot in position and internal memory state $|x, \underline{0}\rangle$. The step operator T for this task is to be such that iteration generates to good accuracy the well known entanglement

$$\phi|x, \underline{0}\rangle \longrightarrow {\sum_y}' c_y|y\rangle|x, \underline{y-x}\rangle \tag{4}$$

over a limited range of y values ($0 \leq y-x < 2^N$ –see below), denoted by the $\prime$ on the $\sum$. Here $|\underline{y-x}\rangle$ is the permanent memory qubit string state corresponding to the lattice distance $y-x$ (number of sites) in one direction only between p and the quantum robot. This equation is easily generalized to the case that the quantum robot is in a wave packet $\theta_{qr} = \sum_x d_x|x\rangle$ of position states to give

$$\phi_p\theta_{qr}|\underline{0}\rangle \longrightarrow {\sum_{x,y}}' c_y d_x|y\rangle|x, \underline{y-x}\rangle. \tag{5}$$

For this task the quantum computer contains two circular qubit quantum registers, one with $N+2$ qubits and the other with $N+1$ qubits, and a head moving on the registers. Both registers accomodate numbers up to $2^N - 1$ with one ternary qubit in each in state $|2\rangle$ serving as an origin. The $N+2$ qubit register is a running memory for the search part counting (with o in state $|mr1\rangle$) and includes a sign qubit. The other serves to hold a permanent copy of the number on the running memory when p is located. Whenever p is found, a computation phase ends the search part by changing the o state to $|ml1\rangle$, carrying out the copying, and subtracting 1 from the running memory to begin the return part. In this part the computation phases subtract 1 from the running memory (no environment observations) until -1 is obtained. The state of o is now changed to $|dn\rangle$ to begin the ballast part. The computation phases subtract 1 from the running memory until $-(2^N - 1)$ is reached when the state of o becomes $|ml >\rangle$. The o state $|mr >\rangle$ is reached if the particle is not located during the search part of the task (i.e. not located in at most $2^N - 1$ iterations of the search action phase). For accurate measurements this phase is entered if $y - x < 0$ or $y - x > 2^N - 1$.

During all action phases the quantum robot moves with the action determined by the state of o. No observations are carried out. The o states $|ml >\rangle$, $|mr >\rangle$ correspond to nonterminating action phases as final task parts. The dynamics of the task can be given schematically by a decision diagram which takes the properties of T_a and T_c into account. This is shown in Figure 1 with details given in the figure caption. The diagram is constructed so that it applies to the subtree emanating from each node of the phase path tree described by Eq. 3. That is, it shows what happens in the subtree based on the overall system state describing the node. This follows from the fact that there is no diagram reference to where the quantum robot or p are on the lattice. $x_{QR} = x_p$? refers only the presence or absence of p at the location of the quantum robot, wherever that is. Also there are no definite duration times associated with either action phases (circles) or with computation phase components (square boxes).

In earlier work T was required to be such that just one phase path in the phase path sum of Eq. 3 contributed. The dispersion in phase path length (t sum) and durations of phases (h sums) was present. Here this restriction will be retained for the computation phase only. For T_c the single path condition is expressed in Eq. 3 by requiring that if j is even, then for each input state $|p(j)\rangle$ to the $j/2$th computation phase, there is a unique output phase path state $|p(j+1)\rangle$. Spreading or quantum dispersion from $|p(j)\rangle$ to $|p(j+1)\rangle$, which is retained, is limited to that given by the h_j sum (and the t sum).

Here the T_a matrix elements $\langle x', l, i|T_a|x, l, 1\rangle$ are required to be local in the sense that their magnitude decreases rapidly as the distance $|x' - x|$ increases. The state of c is denoted by $i = 0, 1$. This is a generalization of the earlier discussion of this example [8] in which only one phase path was included by requiring the T_a matrix element to be 0 unless $x\prime = x$ and $i = 1$, or $x' = x + 1$ and $i = 0$ for $|l\rangle = |mr1\rangle$. For $|l\rangle = |ml1\rangle$ the second condition was replaced by $x' = x - 1$ and $i = 0$.

Phase path dispersion enters here in that the matrix elements $\langle l, x', i|T_a|l, x, 1\rangle$ are $\neq 0$ for different values of $x' - x$. This generates many different action phase output states for each input state with amplitudes determined by sums of products of T_a matrix elements over all paths *within* the action phase. The dependence of the matrix elements on the different final c states $|i\rangle$ shows the action phase contributions to dispersion in the number of phases in a path (t sum) and duration in each action phase (h sum) in the phase path sum of Eq. 3.

B. Accuracy

It is clear from the task description that without further restrictions on T, iteration of T will describe a task and final overall system state that has no relation to the task goal as shown by Eq. 4. To see this let $\Psi(0) = \Theta(0)\phi_p$

represent an initial state with o and c in state $|mr1,0\rangle$ and the quantum robot and p in the lattice position state $|x\rangle\phi_p$ with $\phi_p = \sum_y c_y|y\rangle$. Other initial state values can be obtained from Fig. 1. The probability after k steps that the search part of the task is done and $\underline{n}$ is recorded on the permanent memory is given by

$$P_k(n) = \langle\Psi(k)|P^{st}_{\underline{n}}(1 - P^o_{mr1})|\Psi(k)\rangle = \sum_y |c_y|^2 P_k(n,y) \tag{6}$$

where

$$P_k(n,y) = \langle y, \Theta_k(y)|P^{st}_{\underline{n}}(1 - P^o_{mr1})|y, \Theta_k(y)\rangle. \tag{7}$$

In these equations $\Psi(k) = T^k\Psi(0) = \sum_y c_y\Theta_k(y)|y\rangle$ where $\Theta_k(y)$ is the state of the quantum robot after k steps corresponding to p in state $|y\rangle$. The projection operator $P^{st}_{\underline{n}} = |\underline{n}\rangle\langle\underline{n}|$ where $|\underline{n}\rangle$ is the permanent memory qubit string state corresponding to the number n. The righthand equality of Eq.6, and Eq. 7, express the condition that p is motionless and its state, except for possible entanglement, remains unchanged throughout the task.

The probability $P_k(n,y)$ selects all phase paths in Eq. 3 containing $2n+1$ phases (n action and $n+1$ computation phases) in the completed search part of the task. These paths have in common the fact that in all but the last computation phase p was not at the location of the quantum robot. This corresponds in the search part of the phase path sum to a limitation, for all but the last two phases, to states in which the quantum robot and p are not at the same location. The sum over the output states for the last (nth) action phase of the search part are limited to states in which the quantum robot and p are at the same location y.

The k dependence of $P_k(n)$ enters through the requirement that paths in the phase path sum for $\Theta_k(y)|y\rangle$ contributing to Eq. 7 are those containing n action phases in a completed search part of the task within k steps. The k dependence enters through the h sums of Eq. 3, which express the quantum dispersion in the durations of the different phases. It depends sensitively on the properties of T and on the distance $y - x$. For sufficiently large k the amplitudes of paths, that are still in search parts of the task with $< n+1$ completed computation phases, should be quite small. In this case if T_a is reasonable (e.g. local, etc.) and ϕ is a wave packet localized around some value y_0, then for values of y around y_0, the infinite time limit $P_\infty(n,y)$ should exist and be sensibly equal to $P_k(n,y)$ for large k if $0 \leq y_0 - x < 2^N$.

It may also be the case that for large k the distribution of $P_k(n,y)$, as a function of n may have a peak value and spread around the peak value. However, without further restrictions on T_a, the peak value may have no relation to the distance between p and the quantum robot. One way to remedy this is to require that the matrix elements for T_a have the form $\langle mr1, x', i|T_a|1, x, mr1\rangle = a_i e^{-\alpha(x'-x-1+i)^2}$ where a_i is an i dependent coefficient.. In this case, for large α and k, $P_k(n,y)$ should be peaked around the value $n = y - x$ with a narrow dispersion (provided $0 \leq y - x < 2^N$). In the limit $\alpha,\ k = \infty$ the distance measurement would be completely accurate with no dispersion in the result. In this case $P_\infty(n,y) = \delta_{n,y-x}$ which agrees with Eq. 4. The limit $k \to \infty$ is needed because phase duration dispersion is present in the phase path sum (the exponent is 0 if either $x' = x+1$ and $i = 0$ or $x' = x$ and $i = 1$).

Generalization to the case in which Eq. 5 applies is straightforward. In this case the right hand of Eq. 6 is replaced by $\sum_{x,x',y} d^*_{x'} d_x |c_y|^2 P_k(n,x',x,y)$. The sum is nondiagonal in x and diagonal in y because in this simple example, the quantum robot moves and p is motionless. For large values of $\alpha,\ k$ the values of $P_k(n,x',x,y)$ would be quite small for $x' \neq x$. In the limit $\alpha = \infty$, $P_\infty(n,x',x,y) = \delta_{x'-x}\delta_{n,y-x}$ which agrees with Eq. 5 for $0 \leq y - x < 2^N$.

V. DISCUSSION

Here the existence of a step operator T_c that connects each input path state to a unique output path state in each computation phase was assumed implicitly. The existence of such a T_c follows from the fact that there exists a classical Turing machine step operator whose iterations describe a unique state path *within* each computation phase. A quantum version can be defined that allows spreading along the unique path for each computation phase and thereby introduces the dispersion shown by the h sums of Eq. 3. Generalization to T_c that include sums over different phase states, as was done here for T_a, is left to future work.

The discussion above shows that the description of even a simple distance measurement is relatively complex if specific account is taken of all the steps needed (see Fig. 1) to generate entanglements of the form of Eq. 4. This is based on the representation of numbers as states of quantum registers and the time taken in various pieces of the task (e.g. the $+1$ or -1 operations in the computation phases) and other factors. For the task in question the activities, as shown in Figure 1, represent a task implementation that suggests the existence, for the class of all physical experiments implementable by quantum robots, of a hypothesis [8] similar to the Church Turing Hypothesis [15] for quantum computers [16].

In conclusion it should be stressed that, as an inanimate physical system, the quantum robot knows nothing about counting, or where it is on the lattice at any time, or that it is measuring anything at all. It is simply behaving according to the dynamics specified by T.

ACKNOWLEDGEMENTS

This work is supported by the U.S. Department of Energy, Nuclear Physics Division, under contract W-31-109-ENG-38.

---

[1] P. Shor, in *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society, Los Alamitos, CA 1994), p. 124; Siam Jour. Comput. **26**, 1481 (1997).

[2] L.K.Grover, in *Proceedings of 28th Annual ACM Symposium on Theory of Computing* ACM Press New York 1996, p. 212; Phys. Rev. Letters, **78** 325 (1997); Phys. Rev. Letters, **80** 4329 (1998); G. Brassard, Science **275** 627 (1997).

[3] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, Phys. Rev. Letters **77** 198 (1996); E. Knill and R. Laflamme, Phys. Rev A **55** 900 (1997);P. W. Show, Phys. Rev A**52** R2493 (1995); D. P. DiVincenzo and P. W. Shor, Phys. Rev. Letters **77** 3260 (1996).

[4] E. Knill, R. Laflamme, and W. H. Zurek, Science, **279** 342 (1998).

[5] D. P. DiVincenzo, Science **270** 255 (1995).

[6] N. A. Gershenfeld and I. L. Chuang, Science, **275** 350 (1997); D. G. Cory, A. F. Fahmy, and T. F. Havel, Proc. Natl. Acad. Sci. **94** 1634 (1997).

[7] C. H. Bennett, G. Brassard, E. Bernstein, and U. Vazirani, SIAM Jour. Computing **26** 1510, (1997); *Quantum Oracle Interrogation* W. Van Dam, Los Alamos Archives rept. No. quant- ph/9805006.

[8] P. A. Benioff, Phys Rev. A, to Appear Aug. 1998; Superlattices and Microstructures **23** 407 (1998).

[9] A. Peres and W. Zurek, Amer. Jour. Phys. **50** 807 (1982).

[10] R. Penrose, *The Emperor's New Mind*, Penguin Books, New York, 1991.

[11] H. P. Stapp, *Mind, Matter, and Quantum Mechanics*, Springer Verlag, Berlin 1993.

[12] E. Squires, *Conscious Mind in the Physical World* IOP Publishing, Bristol England, 1990

[13] R.P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals*, McGraw-Hill Book Co. New York 1965.

[14] M. Reed and B. Simon, *Methods of Modern Mathematical Physics*, Vol. 1, Academic Press, New York, 1972, Chap. VIII; I Montavy and G. Münster, *Quantum Fields on a Lattice*, Cambridge University Press, New York, 1994, Chap. 1.

[15] A. Church, Am. Jour. Math. **58** 345 (1936); A. M. Turing, Proc. Lond. Math. Soc. 2 **42**, 230 (1936).

[16] D. Deutsch, Proc. Roy. Soc. (london) A **400** 997 (1985).

Figure 1. Decision Diagram for the Example Task. Task motion is shown by the arrows. The round circles $mr1$, $mr >$, $ml1$, $ml >$, and dn denote action phases. The square boxes denote memory system states (d = running memory and st = permanent memory), questions, and addition ($d = d + 1$) and subtraction ($d = d - 1$) of 1. The boxes and arrows between successive actions show activities of each computation phase. The left hand column shows the task search part dynamics. The central column, with horizontal arrows only, shows memory state changes when p is found, and the righthand column shows the return part dynamics. Ballast activities are shown separately at the bottom. The changes in o system states denoting the end of task parts are not shown as they are easily found from the diagram.