

Debugging Quantum Processes Using Monitoring Measurements

Yangjia Li^{1,2*} and Mingsheng Ying^{2,1†}

¹*State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

²*Centre for Quantum Computation and Intelligent Systems (QCIS),
Faculty of Engineering and Information Technology,
University of Technology, Sydney, NSW 2007, Australia*

(Dated: September 30, 2021)

Since observation on a quantum system may cause the system state collapse, it is usually hard to find a way to monitor a quantum process, which is a quantum system that continuously evolves. We propose a protocol that can debug a quantum process by monitoring, but not disturb the evolution of the system. This protocol consists of an error detector and a debugging strategy. The detector is a projection operator that is orthogonal to the anticipated system state at a sequence of time points, and the strategy is used to specify these time points. As an example, we show how to debug the computational process of quantum search using this protocol. By applying the Skolem–Mahler–Lech theorem in algebraic number theory, we find an algorithm to construct all of the debugging protocols for quantum processes of time independent Hamiltonians.

PACS numbers: 03.67.Ac, 03.65.Ta, 03.67.Pp

I. INTRODUCTION

A major problem in physical implementation of quantum computation is that errors are usually unavoidable in practical situation. To protect the computing process against errors, the method of fault-tolerant quantum computation [1, 2] has been introduced and developed in the last eighteen years. By employing many techniques of quantum error correction [3–5], this method often leads to results in a form of threshold theorem [2, 6]: A quantum computer can be successfully implemented with high probability if each component of the system only fails with probability less than a threshold. The fault-tolerant quantum computation is usually used when the errors are caused by environment noises. The threshold condition is possibly satisfied in this case, as the interaction between the quantum system and the environment may be reduced by other physical techniques, such as [7].

In the present paper, we propose a so-called “debugging” method to deal with another type of errors that are not caused by environment noises but by “bugs”, which mean unknown defects in the physical system itself. The prior techniques for fault-tolerant computation would generally become ineffective for such errors, since the error threshold is mostly broken. For example, suppose a Hadamard gate is by mistake used as a NOT gate in a quantum computer, then this small defect will greatly change the computing result in most cases. A more reasonable strategy here is to find the nature and exact position of this defect, and then repair it. To this end, quantum measurements should be applied to monitoring the computing process so that errors can be detected as soon as possible after the component with

bugs being executed. Remarkably, a debugging method like this plays an indispensable role and attracts intense studies [8] in the implementation of classical computing systems.

Unfortunately, due to the fundamental difference between the physical behavior of quantum measurements and that of classical ones, debugging for quantum systems is much more difficult than for classical systems, and thus classical debugging method does not work in the quantum scenario. Specifically, in quantum mechanics, observation of a quantum system would make the system state collapse. This interaction between observing apparatus and quantum systems on the one hand allows quantum measurements to drive target systems as quantum operations [5, 9], in applications like teleportation [10], entanglement distillation [11], control of quantum systems [12], and one-way quantum computing [13]; but on the other hand, it makes many tasks much harder than in the classical world, particularly when quantum measurements are used to extract (classical) information of given systems. The well known indistinguishability between nonorthogonal states can somehow be seen as a simple example. The quantum debugging task considered here is actually another instance, where measurements monitor the system state for possible errors. This can be easily done for a classical process, because the trajectory of a classical system is unchanged by measurements. However, a problem in monitoring a quantum process is that once the system had been measured, the system state may be disturbed and then be useless for further processing. This problem has been demonstrated to be very serious in the quantum Zeno effect [14], that a quantum process can be completely obstructed by continuing measuring. Therefore, similar tasks are usually achieved by quantum tomography techniques [5, 15] in the literature, in which the system state is measured only once to keep the outcome faithful, but instead, a large number

* liyj04@mails.tsinghua.edu.cn

† Mingsheng.Ying@uts.edu.au

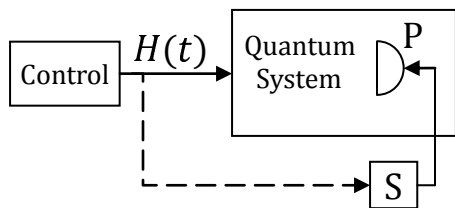


FIG. 1. The classical control information about $H(t)$ is sending to S during the execution of the process. Then at any time t , S can decide whether or not $P|\psi_t\rangle = 0$ according to the control history. And if it will drive P to detect possible errors.

of copies of the process are required.

In the debugging method proposed in this paper, quantum measurements are used in a different way: they are constantly taken to monitor a quantum process but without disturbances on the system state, until an error is detected. A basic scheme is described as follows. Consider a quantum system that is established to run some computing process. It is designed to be in state $|\psi_0\rangle$ initially, and then evolves under the controlled Hamiltonian $H(t)$. In this way, the trajectory $\{|\psi_t\rangle\}$ of the system state would be as anticipated. The time for the whole process is considered to be infinite, as it is usually much longer than the time for a single component (like a gate) acting. Now suppose a bug of the system will be involved in the process at time t' , then the system Hamiltonian will not truly be $H(t)$ for $t \geq t'$ in the practical execution. This causes errors in system state, so we write ρ_t for the density operator of the actual state at time t . To debug the process, we need to find a projection operator $P \neq 0$ of the system and a sequence of time points t_1, t_2, \dots ($t_n \rightarrow \infty$), such that $P|\psi_{t_n}\rangle = 0$ for all n . The condition of $P|\psi_t\rangle = 0$ means that nothing can be detected by P if the system state is $|\psi_t\rangle$ as anticipated. We monitor the process at time t_1, t_2, \dots , using a measurement apparatus formalized by P . This measurement is called a monitoring measurement. Then with probability $\text{tr}(P\rho_{t_n})$ the error state would be detected at time t_n . If it really happens, then an error is detected in the state. In this case, t' is more likely in $[t_{n-1}, t_n]$ and the relevant components should be carefully checked. Practically, the time points t_1, t_2, \dots are determined by a classical program S . Then the debugging protocol is visualized as FIG. 1. Obviously, the key step of debugging a process is to find the required projection operator P . The condition of $P|\psi_t\rangle = 0$ guarantees that the anticipated process is not disturbed by P . On the other hand, it implies that the protocol is conclusive; i.e., no errors would be reported when the process runs correctly.

The aim of this paper is to develop the debugging method for quantum systems outlined above. The paper is organized as follows. In Sec. II, we first consider an example debugging protocol for quantum search algorithm. After that, we propose a general debugging scheme and show that it can be reduced to a simpler scheme described

as in FIG 1. Then we formally define this simplified debugging protocols in the case of discrete time evolution. In Sec. III, we completely solve the debugging problem for quantum processes with time independent Hamiltonians. More precisely, we find an algorithm to construct all of the debugging protocols for this kind of quantum processes by employing the celebrated Skolem–Mahler–Lech theorem. A brief conclusion is drawn in Sec. IV.

II. DEBUGGING PROTOCOLS

A. An Example

To show how can the debugging method be truly applied, let us first consider a simple example — debugging for the computational process of quantum search [16]. Here, we adopt the description of the Grover algorithm in [5]. The quantum computer consists of n qubits with $|0\rangle^{\otimes n}$ as the initial state (for simplicity, we omit the auxiliary qubits of the oracle). A black box oracle O of form

$$O = I_2^{\otimes n} - 2|x\rangle\langle x|$$

is provided as input, where $x \in \{0, 1, \dots, 2^n - 1\}$ is the index we want to find. The computer first applies $H_2^{\otimes n}$, and then successively applies the Grover iteration

$$G = (2|\psi_0\rangle\langle\psi_0| - I_2^{\otimes n})O$$

for $O(\sqrt{2^n})$ times, where

$$|\psi_0\rangle = \sum_{k=0}^{2^n-1} |k\rangle / \sqrt{2^n}.$$

At last, x can be obtained with probability $O(1)$ by measurement in the computational basis $\{|0\rangle, |1\rangle\}$ on each qubit. Here, we use I_2 and H_2 to denote the identity and Hadamard gates, respectively.

To debug this process, we note that the system state immediately after each Grover iteration should be always in the two-dimensional subspace $\text{span}\{|x\rangle, |\xi\rangle\}$, where $|\xi\rangle = \sum_{k \neq x} |k\rangle / \sqrt{2^n - 1}$. So, we can use a measurement apparatus formalized by $P = I_2^{\otimes n} - |x\rangle\langle x| - |\xi\rangle\langle\xi|$ to detect errors. The protocol is as follows: randomly choose an integer x and provide the corresponding oracle at the beginning, and then execute the algorithm. Immediately after each Grover iteration G , take the monitoring measurement P to detect errors. If an error system state is detected at some time point, then the debugging protocol stops the process and reports this error. Now we particularly discuss the following two kinds of bugs:

1. The system was not initialized. We write ρ for the density operator of the initial system state and write f for the fidelity of ρ and $|0\rangle^{\otimes n}$. Then it is easy to verify that with probability $(2^n - 2)(1 - f^2)/(2^n - 1)$ an error can be detected just by the first measurement of P .

- The Grover iterator was implemented with some bugs, so it is not G but some unitary operator G' . In most cases, the two subspaces $\text{span}\{G'|x\rangle, G'|\xi\rangle\}$ and $\text{span}\{G|x\rangle, G|\xi\rangle\} = \text{span}\{|x\rangle, |\xi\rangle\}$ have no common state. So $|\langle x|G'|\psi\rangle|^2 + |\langle \xi|G'|\psi\rangle|^2 < 1$ for all $|\psi\rangle \in \text{span}\{|x\rangle, |\xi\rangle\}$. We write q for the maximal value of all $|\langle x|G'|\psi\rangle|^2 + |\langle \xi|G'|\psi\rangle|^2$. Then at each measurement of P , an error will be detected with a positive probability at least $1 - q > 0$.

Two advantages of the quantum debugging method are demonstrated in this example: (1) An error may be detected soon after the bugs involved. So, the process can be just partly executed and a lot of time would be saved; (2) A single execution of the process is usually sufficient to detect an error, whereas a large number of copies of the process are needed in other approaches.

B. A General Debugging Protocol

We now consider a general scheme of debugging protocols for quantum processes, in which quantum measurements are in the most general form, and different measurements can be used at different time points to detect errors. First, we impose a *compatibility* constraint to each monitoring measurement, such that the target system state keeps unchanged under its action. Formally, the compatibility can be stated as follows: let $|\psi\rangle$ be a state and $\mathcal{M} = \{M_1, M_2, \dots, M_k\}$ be a measurement. We say that M is compatible with $|\psi\rangle$ if for all i , $M_i|\psi\rangle$ is essentially the same as $|\psi\rangle$ or vanish; that is, $|\psi\rangle$ is an eigenstate of every measurement operator of \mathcal{M} :

$$\forall i \exists \lambda_i \text{ s.t } M_i|\psi\rangle = \lambda_i|\psi\rangle. \quad (1)$$

In fact, this constraint simulates the physical behavior of a classical measurement: The states of a classical system can be thought of as an orthonormal basis $\{|i\rangle\}_{i=1}^k$, and we consider a classical measurement $\mathcal{M} = \{M_1, \dots, M_k\}$ with $M_i = |i\rangle\langle i|$. Then the compatibility is automatically satisfied: $M_i|j\rangle = \delta_{ij}|j\rangle$ for each i and j .

A general protocol for debugging a quantum process using monitoring quantum measurements consists of three steps:

- Set a sequence of breakpoints at time t_1, t_2, \dots during the process;
- Execute the process, and at each breakpoint of time t_n , insert a measurement $\mathcal{M}_{t_n} = \{M_{t_n1}, M_{t_n2}, \dots, M_{t_nk_n}\}$ that is compatible with the anticipated system state $|\psi_{t_n}\rangle$. We write $E(\mathcal{M}_{t_n}) = \{i|M_{t_ni}|\psi_{t_n}\rangle = 0\}$ for the outcomes i that should not occur at time t_n if the process behaves as anticipated;
- An error is detected if the measurement outcome at t_n is some element $i \in E(\mathcal{M}_{t_n})$. In this case we stop the execution and report i and t_n to specify the error type and the error position, respectively.

We can actually simplify this general debugging scheme without loss of generality. First, we show that at each breakpoint of time t , the general quantum measurement $\mathcal{M}_t = \{M_{ti}\}$ can be replaced with the two-outcome POVM $\{I - E_t, E_t\}$, where $E_t = \sum_{i \in E(\mathcal{M}_t)} M_{ti}^\dagger M_{ti}$ is used to indicate errors and $I - E_t$ indicates correctness. Here I is the identity operator of the system. In fact, this POVM performs mostly the same as \mathcal{M}_t : They are both compatible with $|\psi_t\rangle$ and detect errors with the same probability $\text{tr}(E_t\rho_t)$. Here we denote by ρ_t the system state with errors. The only disadvantage of such replacement is that different error types i in $E(\mathcal{M}_t)$ are not distinguished. However, this would not be a problem because after an error being detected, the type can be specified by further measurement. Moreover, it is even better to use the projective measurement $\{I - P_t, P_t\}$ with P_t being the projection operator into the support of E_t . This is because the measurement also satisfies the compatibility, and it detect errors with probability $\text{tr}(P_t\rho_t) \geq \text{tr}(E_t\rho_t)$. Therefore, it suffices to detect errors using monitoring measurements formalized by projection operators P_t . We will call them error detectors in what follows.

Secondly, we assert that all of the error detectors P_t should be chosen only from a finite set; otherwise, the protocol would be useless. The reason is that if infinitely many detectors are used, then to decide which one is chosen at a breakpoint, the amount of information we needed would become infinite. A specific instance is helpful to understand this situation: At each breakpoint of time t , we simply use $P_t = \psi_t^\perp$ as the error detector. Obviously, it is compatible with $|\psi_t\rangle$ and any error of this system state can be detected using it. However, to construct this detector we need the complete information of $|\psi_t\rangle$ by classical computation; namely, the debugging protocol requires a classical simulation of the quantum process, which is clearly unreasonable. So, the requirement of finiteness is crucial for effective debugging protocols. We write all the detectors as P_1, P_2, \dots, P_k . Then there is a strategy S for the protocol to call one of them at each breakpoint. Now we can divide the strategy S into k parts S_1, S_2, \dots, S_k , where S_i is a strategy that only call P_i at corresponding breakpoints and keeps silent at the others. Then the original debugging protocol can be decomposed as k protocols $(P_i, S_i)(i = 1, 2, \dots, k)$, each of which monitors the process at a part of breakpoints. In particular, some of the protocols will constantly work at an infinite subsequence of the breakpoints.

Therefore, we only need to investigate the debugging protocols of such a form: it consists of an error detector P and a strategy S ; at a sequence of time points specified by S , P is taken to detect possible errors of the system state. We note that this simplified protocol is exactly that visualised in FIG. 1. If all protocols in this scheme can be found for a given quantum process, then a general debugging task can be achieved by a simple combination of them, with certain further analysis about the detected errors.

C. Discrete Time Evolution

Since an error detector P is discretely taken in the debugging described above, it is reasonable to consider the discrete time evolution of the system. Specifically, we assume that the compatibility constraint is only checked by strategy S at given points t_0, t_1, \dots of time. Then it suffices to considering the corresponding states $|\psi_{t_0}\rangle, |\psi_{t_1}\rangle, \dots$, and the state transformations between them, which are formalized by unitary operations. In this way, the design of a quantum process can be depicted as

$$|\psi_{t_0}\rangle \xrightarrow{U_{\alpha_1}} |\psi_{t_1}\rangle \xrightarrow{U_{\alpha_2}} |\psi_{t_2}\rangle \xrightarrow{U_{\alpha_3}} \dots,$$

where $|\psi_{t_n}\rangle = U_{\alpha_n}|\psi_{t_{n-1}}\rangle$ for every $n \geq 1$, and U_{α_n} describes the evolution of the system from time t_{n-1} to t_n . For realizability, we can assume that all of these unitary operators can be chosen from a finite set $\{U_1, \dots, U_m\}$. Then we have $\alpha_n \in \{1, \dots, m\}$ for every $n = 1, 2, \dots$. Obviously, a circuit model of quantum computation can be seen as a quantum process like this, where U_1, \dots, U_m are the gates in the circuit. Quantum walk [17] can be considered as another example of quantum processes in this form.

Now we rigorously define the debugging protocol (P, S) for quantum processes formulated by such a system. An error detector P is a projection operator in the state Hilbert space \mathcal{H} , and a strategy S is a function that to each finite sequence $s = \alpha_1\alpha_2 \dots \alpha_n$ of indices in $\{1, \dots, m\}$, assigns a result of “yes” or “no”. Intuitively, $S(s) = \text{“yes”}$ (resp. “no”) means that P is (resp. not) used to detect errors immediately after the execution of the action sequence $U_{\alpha_1}, U_{\alpha_2}, \dots, U_{\alpha_n}$. For simplicity, we write $U_s = U_{\alpha_n} \dots U_{\alpha_2} U_{\alpha_1}$ for the composition of the corresponding unitary actions. To warrant the protocol actually realizable, the following three conditions are necessary:

1. (Compatibility) $S(s) = \text{“yes”}$ implies $PU_s|\psi_{t_0}\rangle = 0$.
2. (Computability) A classical algorithm can be found to compute S .
3. (Liveness) For any infinite sequence $\alpha_1\alpha_2 \dots$ of indices $1, \dots, m$ there are infinitely many n 's such that $S(\alpha_1\alpha_2 \dots \alpha_n) = \text{“yes”}$.

The first two conditions are easy to understand. The liveness comes from the fact that P should constantly be applied in the process represented by $\alpha_1\alpha_2 \dots$, so that bugs involved at any time could be detected.

Based on the above definition of debugging protocol, a debugging problem can be formally stated as follows:

- Given an initial state $|\psi_{t_0}\rangle$ and a set of unitary operations U_1, U_2, \dots, U_m that describe the discrete-time evolution of a quantum process, how can we find all the protocols (P, S) satisfying Compatibility, Computability and Liveness?

III. DEBUGGING FOR TIME-INDEPENDENT HAMILTONIANS

A. A Basic Theorem

We now solve the debugging problem for the case where the designed Hamiltonian is time independent. Specifically, our solution consists of the following three steps:

1. We find a method to check whether or not a given projection operator P can be used as an error detector;
2. For each eligible P , we show that a strategy S can be constructed as a periodic function;
3. We present a procedure that can compute all the debugging protocols (P, S) for any given process.

Let \mathcal{H} be the state Hilbert space of the system, and H the system Hamiltonian which is time independent. To define debugging protocols (P, S) , we consider the discrete time evolution of the system between a sequence of time points $0, \Delta t, 2\Delta t, \dots$, where Δt is a fixed period of time which can be appropriately chosen in practice. Then at time $n\Delta t$, the anticipated system state is $|\psi_n\rangle = U^n|\psi_0\rangle$, where $|\psi_0\rangle$ is the initial state and $U = \exp(-iH\Delta t/\hbar)$ is the unitary transformation of time evolution in a single period. As defined in Subsec. II-C, a debugging protocol for this system consists of an error detector P which is a projection operator of \mathcal{H} , and a strategy S which is a function specifying (by assigning “yes”) an infinite sequence of integers i_1, i_2, \dots such that $PU^{i_n}|\psi_0\rangle = 0$ for all n . Our task is to find the detector P and the strategy S .

Obviously, a necessary condition of P being an error detector is that $PU^n|\psi_0\rangle = 0$ for infinitely many n . To investigate how this condition can be satisfied, we need the following theorem:

Theorem 1. Let $|\psi_0\rangle$ be a vector, U a unitary operator and P a projection operator in a finite dimensional space \mathcal{H} . If $Z = \{n|PU^n|\psi_0\rangle = 0\}$ is an infinite set, then an arithmetic progression $\{pn + r|n = 0, 1, \dots\}$ can be algorithmically found in Z .

The proof of Theorem 1 is postponed to next subsection. Here we see how this theorem can be used in our investigation of a debugging protocol (P, S) . First, the infiniteness condition of Z can be checked, as it is equivalent to the existence of the arithmetic progression. Second, this condition is not only necessary but also sufficient for P being an error detector. In fact, if it holds for P , then we can construct a strategy S as a periodic function that assigns “yes” to the integers $pn+r$, $n = 0, 1, \dots$, and “no” to the others. Moreover, by making the arithmetic progression $\{pn + r|n = 0, 1, \dots\}$ exist in Z , we have a procedure to compute all the error detectors P . Such a procedure will be carefully described in Subsec. III-C based on the proof of the theorem.

B. Proof of Theorem 1

A key step in the proof of Theorem 1 is to explore the implication of the infiniteness of Z . For this purpose, we employ some techniques from the previous research on the famous Skolem's problem [18]. Consider a linear recurrent sequence $\{a_n\}_{n=0}^{\infty}$, which satisfies the linear recurrence relation:

$$a_{n+d} = c_{d-1}a_{n+d-1} + c_{d-2}a_{n+d-2} + \cdots + c_0a_n \quad (2)$$

for all $n \geq 0$. Let $Z = \{n | a_n = 0\}$ be the set of indices of null elements of $\{a_n\}$. A way relating the above linear recurrent sequence to the behavior of a quantum system is putting $a_n = \langle \phi | M^n | \psi \rangle$ for two quantum states $|\phi\rangle, |\psi\rangle$ and a quantum operation M of a d dimensional quantum system. Remarkably, this technique has already been successfully used to solve several important problems in quantum information theory. For example, the condition $\langle \phi | M^n | \psi \rangle = 0$ is interpreted as the acceptance condition of finite quantum automata in [19] for M being a unitary operator, and as the occurrence of specific quantum measurement outcomes in [20] for M being a measurement operator, respectively. The decision problems considered in [19, 20] are similar to the Skolem's emptiness problem [21]. What we need in the proof of our result is the following [22]:

Theorem 2 (Skolem–Mahler–Lech). In a field of characteristic 0, let a sequence $\{a_n\}_{n=0}^{\infty}$ satisfy a recurrence relation of form Eq. (2), then the set Z of indices of null elements of this sequence is semi-linear, namely, is a union of a finite set and finitely many arithmetic progressions.

To apply this theorem to our problem, we decompose $P = \sum |\phi_i\rangle\langle\phi_i|$, where states $|\phi_i\rangle$ form an orthonormal basis of the image space of P . Let $\lambda^d - c_{d-1}\lambda^{d-1} - c_{d-2}\lambda^{d-2} - \cdots - c_0$ be the characteristic polynomial of U . Then for each $|\phi_i\rangle$, we can invoke Theorem 2 for $a_n = \langle \phi_i | U^n | \psi_0 \rangle$ and assert that the set $Z_i = \{n | \langle \phi_i | U^n | \psi_0 \rangle = 0\}$ is semi-linear. Furthermore, we see that $Z = \bigcap Z_i$ is also semi-linear. Thus, the infiniteness of Z in Theorem 1 actually implies that it contains at least one arithmetic progression.

There is still a gap between the existence of the arithmetic progression in Theorem 1 and its algorithmic construction. Here we further present an algorithm to find p and r such that $PU^{pn+r}|\psi_0\rangle = 0$ for all $n = 0, 1, \dots$. Of course we should assume that all operators and states are represented by matrices and vectors of rational complex numbers.

Finding number p : We can algorithmically find a positive integer p satisfying the following condition:

- for any two eigenvalues λ and μ of U , $(\lambda/\mu)^p = 1$ provided $(\lambda/\mu)^n = 1$ for some integer n .

Indeed, it suffices to find the smallest positive integer n satisfying $(\lambda/\mu)^n = 1$ for each fixed pair of λ, μ , and then p can be chosen as the least common multiple of

all these n . We note that all roots of the characteristic polynomial $f(x)$ of $U \otimes U^\dagger$ are exactly all quotients λ/μ of two eigenvalues of U . Moreover, for each quotient λ/μ , if n is the smallest positive integer number satisfying $(\lambda/\mu)^n = 1$, then λ/μ should be a root of the n th cyclotomic polynomial $\Phi_n(x)$, and $\Phi_n(x)$ should be a divisor of $f(x)$ since $\Phi_n(x)$ is irreducible in the field of rational numbers. Therefore, all of such n can be obtained by checking whether or not $\Phi_n(x) | f(x)$.

Moreover, we prove that the number p enjoys an property: for any subspace K of \mathcal{H} , $U^p K = K$ provided $U^n K = K$ for some integer n . We observe that $U^n K = K$ if and only if a set of eigenvectors of U^n forms a basis of K . From this observation, it suffices to prove that any eigenvector of U^n is an eigenvector of U^p . More generally, we show that any eigenspace E of U^n is included in some eigenspace of U^p . We note that all eigenvectors of U are eigenvectors of U^n , so we can choose a set of eigenvectors of U to form a basis B of E . Consider any two of these vectors, written as $|\psi\rangle$ and $|\phi\rangle$, and we write λ and μ , respectively, for the corresponding eigenvalues of U . Then we have $(\lambda/\mu)^n = 1$, and according to our choice of p , $(\lambda/\mu)^p = 1$. So $|\psi\rangle$ and $|\phi\rangle$ are in the same eigenspace of U^p . As these two states are arbitrarily chosen, it implies that all of the vectors in B are in the same eigenspace of U^p . Thus E is included in it.

Finding number r : Let $K = \{|\psi\rangle | P|\psi\rangle = 0\}$ be the kernel space of P . For any integer q , we write K_q for the maximal subspace of K satisfying $U^q K_q = K_q$. Then K_q can be calculated by the iteration $K_q \leftarrow K_q \cap U^q K_q$, putting $K_q \leftarrow K$ initially. On the other hand, we show that

$$K_q = \{|\psi\rangle \in K | U^{nq}|\psi\rangle \in K \text{ for all integer } n \geq 0\}. \quad (3)$$

First, for any state $|\psi\rangle \in K_q$, one can easily verify from the definition of K_q that $U^{nq}|\psi\rangle \in K_q \subseteq K$ for all n . Secondly, if some state $|\psi\rangle$ satisfies $U^{nq}|\psi\rangle \in K$ for all n , then we consider the subspace of K : $K' = \text{span}\{U^{nq}|\psi\rangle | n = 0, 1, \dots\}$. We have $U^q K' = K'$, and thus $|\psi\rangle \in K' \subseteq K_q$ from the maximality of K_q . Therefore, Eq. (3) holds.

To make $U^{pn+r}|\psi_0\rangle = 0$ for all $n \geq 0$, it suffices to calculate K_p and then find r from $\{0, 1, \dots, p-1\}$ such that $U^r|\psi_0\rangle \in K_p$. Now we only need to prove the following claim:

- Whenever there exists an arithmetic progression $\{an + b | n = 0, 1, \dots\}$ in Z , the number r can be found as above.

In fact, by Eq. (3), $\{an + b | n = 0, 1, \dots\} \subseteq Z$ means that $U^b|\psi_0\rangle \in K_a$. We note that $U^a K_a = K_a$ implies $U^p K_a = K_a$ by the property of p stated above. Thus, $K_a \subseteq K_p$ due to the maximality of K_p . So we have $U^b|\psi_0\rangle \in K_p$. If we put $r = b - cp \in \{0, 1, \dots, p-1\}$ as the remainder of b divided by p , then $U^r|\psi_0\rangle \in U^{-cp} K_p = K_p$. So r can be obtained in the algorithm. This completes the proof of Theorem 1.

C. Construction of the Debugging Protocols

Now we can construct all debugging protocols (P, S) for a given process using the proof of Theorem 1. A necessary and sufficient condition of error detectors P is that $PU^{pn+r}|\psi_0\rangle = 0$ ($n \geq 0$) for the integer p and some $r \in \{0, 1, \dots, p-1\}$. So the construction of (P, S) is achieved in four steps:

1. Compute the number p from the given unitary operator U . An algorithm for finding p was already presented in the proof of Theorem 1.
2. Arbitrarily choose a number $r \in \{0, 1, \dots, p-1\}$, and compute the subspace

$$V_r = \text{span}\{U^{pn+r}|\psi_0\rangle | n = 0, 1, \dots, d-1\},$$

where d is the dimension of the system.

3. P can be chosen as any projection operator satisfying $PV_r = 0$. In particular, we choose it as the one with image space V_r^\perp , since it is of the maximal rank and thus can detect as many as possible errors.
4. S is constructed as the periodic function that specifies the arithmetic progression $\{pn + r | n = 0, 1, \dots\}$.

As an instance, we show how the above procedure can be used to construct a debugging protocol for the quantum search process in Subsec. II-A. The computational process of quantum search can be formalized in our model: the Hilbert space \mathcal{H} is of dimension $N = 2^n$, the initial state is $|\psi_0\rangle = \sum_{k=0}^{N-1} |k\rangle / \sqrt{N}$ and the unitary transformation is

$$G = (2|\psi_0\rangle\langle\psi_0| - I_2^{\otimes n})(I_2^{\otimes n} - |x\rangle\langle x|),$$

where $x \in \{0, 1, \dots, N-1\}$ is a given integer. Then the number p , number r , and projection operator P are determined as follows:

1. To obtain the number p , we calculate the characteristic polynomial of G that is

$$(\lambda - 1)^{N-2}(\lambda^2 + 2(1 - 2/N)\lambda + 1).$$

We only consider the case of $N > 4$. It is easy to verify that for any two eigenvalues λ, μ of G , if $(\lambda/\mu)^n = 1$ for some n then $\lambda = \mu$. So we have $p = 1$.

2. Now $r \in \{0, 1, \dots, p-1\}$ can only be 0 because $p = 1$. Then

$$\begin{aligned} V_0 &= \text{span}\{U^n|\psi_0\rangle | n = 0, 1, \dots, N-1\} \\ &= \text{span}\{|x\rangle, |\xi\rangle\}, \end{aligned}$$

where $|\xi\rangle = \sum_{k \neq x} |k\rangle / \sqrt{N-1}$.

3. We choose $P = I_2^{\otimes n} - |x\rangle\langle x| - |\xi\rangle\langle\xi|$ to make the condition $PV_0 = 0$ be satisfied.

As $p = 1$ and $r = 0$, P is applied immediately after each action of G . We see that this protocol constructed by the procedure presented in this subsection is exactly that given in Subsection II-A.

IV. CONCLUSION

In this paper, we proposed a scheme for debugging a quantum process, in which quantum measurements are used to monitor the system without disturbances on its behaviour. We discovered a procedure to construct all debugging protocols in this scheme for quantum processes with time independent Hamiltonians. However, the problem of debugging quantum processes is still open for the case of time dependent Hamiltonians.

ACKNOWLEDGEMENT

We are grateful to Runyao Duan, Yuan Feng and Nengkun Yu for useful discussions. This work was partly supported by the Australian Research Council (Grant No: DP110103473 and DP130102764).

-
- [1] P. W. Shor, in Proc. 37th Annual Symposium on Foundations of Computer Science, 56-65 (IEEE Press, Los Alamitos, 1996).
 - [2] A. Y. Kitaev, Russ. Math. Surv. **52**, 1191 (1997).
 - [3] P. W. Shor, Phys. Rev. A **52**, 2493 (1995).
 - [4] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
 - [5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
 - [6] E. Knill, R. Laflamme, and W. H. Zurek, Science **279**, 342 (1998).
 - [7] J. Zhang, A. M. Souza, F. D. Brandao, and D. Suter, Phys. Rev. Lett. **112**, 050502 (2014).
 - [8] M. Leucker and C. Schallhart, Journal of Logic and Algebraic Programming, **78**, 293 (2009); G. J. Myers, *The Art of Software Testing* (John Wiley and Sons, Inc. 1979).
 - [9] R. Y. Duan, Y. Feng, and M. S. Ying, Phys. Rev. Lett. **103**, 210501 (2009).
 - [10] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Phys. Rev. Lett. **70**, 1895 (1993).
 - [11] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Phys. Rev. Lett. **76**, 722 (1996).
 - [12] C. Altafini, and F. Ticozzi, IEEE Transactions on Automatic Control, **57**, 1898 (2012).

- [13] R. Raussendorf and H. J. Briegel, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [14] B. Misra and E. C. G. Sudarshan, *J. Math. Phys.* **18**, 756 (1977).
- [15] J. F. Poyatos, J. I. Cirac, and P. Zoller, *Phys. Rev. Lett.* **78**, 390 (1997); I. L. Chuang and M. A. Nielsen, *J. Mod. Opt.* **44**, 2455 (1997).
- [16] L. Grover, in *Proc. 28th Annual ACM Symposium on the Theory of Computing*, 212-219 (ACM Press, New York, 1996).
- [17] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous, in *Proc. 33rd Annual ACM Symposium on the Theory of Computing*, 37-49 (ACM Press, New York, 2001).
- [18] T. Skolem, in *Proc. 8th Congress of Scandinavian Mathematicians*, 163-188, (Stockholm, 1934).
- [19] V. D. Blondel, E. Jeandel, P. Koiran and N. Portier, *SIAM J. Comput.* **34**, 1464 (2005).
- [20] J. Eisert, M. P. Müller and C. Gogolin, *Phys. Rev. Lett.* **108**, 260501 (2012).
- [21] V. Halava, T. Harju, M. Hirvensalo and J. Karhumäki, *TUCS Technical Report*, 683 (2005).
- [22] C. Lech, *Ark. Mat.* **2**, 417-421 (1953).