Near-time-optimal control for quantum systems

Qi-Ming Chen,1 Re-Bing Wu,1,* Tian-Ming Zhang,1 and Herschel Rabitz2
1Department of Automation, Tsinghua University, Beijing, 100084, China
2Department of Chemistry, Princeton University, Princeton, New Jersey 08544, USA
(Rceived 29 July 2015; published 16 December 2015)

For a quantum system controlled by an external field, time-optimal control is referred to as the shortest-time-duration control that can still permit maximizing an objective function $J$, which is especially a desirable goal for engineering quantum dynamics against decoherence effects. However, since rigorously finding a time-optimal control is usually very difficult and in many circumstances the control is only required to be sufficiently short and precise, one can design algorithms seeking such suboptimal control solutions for much reduced computational effort. In this paper, we propose an iterative algorithm for finding near-time-optimal control in a high level set (i.e., the set of controls that achieves the same value of $J$) that can be arbitrarily close to the global optima. The algorithm proceeds seeking to decrease the time duration $T$ while the value of $J$ remains invariant, until $J$ leaves the level-set value; the deviation of $J$ due to numerical errors is corrected by gradient climbing that brings the search back to the level-set $J$ value. Since the level set is very close to the maximum value of $J$, the resulting control solution is nearly time optimal with manageable precision. Numerical examples demonstrate the effectiveness and general applicability of the algorithm.

DOI: 10.1103/PhysRevA.92.063415  PACS number(s): 32.80.Qk, 02.30.Yy, 07.05.Dz

I. INTRODUCTION

When quantum systems are complex or when they are exposed to unwanted environments, control theory provides powerful tools for improving the performance of quantum operations [1]. Over the past few decades, considerable successes have been achieved in the control of atomic, molecular, optical, and solid-state quantum systems [2–6]. Since quantum operations are effective only within the coherence time, the control needs to either prolong the coherent dynamics via dynamical decoupling [7,8] or feedback [9] strategies or be performed as rapidly as possible [10–12]; the latter circumstance poses the quantum time-optimal control problem. Based on Pontryagin’s maximum principle [13–16], geometric analysis shows that the minimal time is mainly determined by the evolution speed of the drift Hamiltonian when the controls are unbounded. Such time-optimal gate control solutions have been found for one-qubit, two-qubit, and some special three-qubit systems [17,18]. Cases with bounded controls are far more complex, and only for several special cases have analytical [19–27] or partially analytical [28,29] solutions been found.

Under most circumstances, quantum time-optimal control problems can be numerically solved by minimizing the time duration while maintaining the objective function $J$ at the maximum value (e.g., gate fidelity). To first maximize the objective function $J$ with fixed $T$, the genetic algorithm [30], the monotonic iterative algorithm [31], the Krotov algorithm [32], the gradient algorithm [18], and the hybrid algorithm [33] are introduced for the search of optimal controls. To further shorten the control time, a natural approach is to gradually decrease $T$ and apply the algorithm that maximizes $J$ for each fixed $T$ until the desired control precision can no longer be attained [18,34]. One can also take the time duration as a control variable and design “free-time and fixed-end-point” algorithms, by which the time duration and the control error can be decreased [35–37] in the presence of trade-offs with other penalty functions (e.g., field fluence).

Although the optimization of $J$ itself with fixed $T$ is generally very efficient [38–41], the search for a precise time-optimal control is much more expensive due to its biobjective nature and the potential existence of traps at reduced values of $J$ [33,34]. However, as will be shown in this paper, the search for minimum-time solutions is relatively easy if $J$ is allowed to be slightly lower than its maximal value. Such a control is nearly time optimal because both $J$ and the corresponding optimal time are very close to their ideal optimal values. From an engineering point of view, this solution is likely acceptable because control precision is usually limited by the tomography and pulse-shaping techniques (e.g., the fidelity of tomography experiments is seldom above 99%), and it makes no substantial difference when the control time is slightly prolonged. Therefore, it is worth designing algorithms for finding such control protocols that can be computationally much cheaper.

This paper will present a two-stage level-set-based algorithm (LS algorithm) seeking near-time-optimal control solutions. The algorithm will first climb to a level set, which is referred to as the set of control functions corresponding to a fixed value of $J$, then find a highly efficient path toward minimum-time control in this level set until a satisfactory solution is found. The paper is organized as follows. In Sec. II, we define the time-optimal and near-time-optimal control problems for quantum systems. In Sec. III, we present the strategy and details of the algorithm, which is then illustrated by examples in Sec. IV. Finally, we draw conclusions in Sec. V.

II. TIME-OPTIMAL CONTROL PROBLEMS

Consider an $N$-dimensional quantum control system whose unitary propagator $U(t)$ obeys the following Schrödinger
equation:
\[
\frac{d}{dt} U(t) = -i \left[ H_0 + \sum_{k=1}^{m} u_k(t) H_k \right] U(t),
\]
where the free Hamiltonian $H_0$ and the control Hamiltonians $H_k$ are Hermitian operators in the underlying Hilbert space (the common constant $\hbar$ has been absorbed into the Hamiltonians) and the system is manipulated by the multiple control functions $u_1(t), u_2(t), \ldots, u_m(t)$, which act during the time interval $t \in [0, T]$.

Time-optimal control problems are always associated with an objective $J$ as a function of the controls. For example, in quantum information processing, we have
\[
J_1 = N^{-1} \text{Re} \text{Tr}[W U(T)],
\]
where $\text{Re}(\cdot)$ denotes the real part of a complex number, for implementing a target unitary evolution $W$ at time $T$ or
\[
J_2 = \text{Tr}[U(t) \rho_0 U(t)^\dagger O]
\]
for maximizing the expectation value of the quantum observable $O$, where $\rho_0$ is the initial density matrix.

In numerical simulations, we discretize the whole time interval $[0, T]$ into $M$ steps at points
\[
t_j = T \frac{j}{M}, \quad j = 1, \ldots, M,
\]
and take the values of $u_{kj} \triangleq u_k(t_j)$ at these time instances as the control variables:
\[
\mathbf{u} \triangleq [u_{11}, \ldots, u_{1M}, \ldots, u_{m1}, \ldots, u_{mM}]^T.
\]

Considering $T$ as an additional control variable, the objective $J$ is a function of the $(mM + 1)$-dimensional vector $\mathbf{x} = [T, \mathbf{u}]$.

For convenience, we denote the set of control vectors
\[
\mathcal{L}(J = J_0) = \{ \mathbf{x} : J(\mathbf{x}) = J_0, T \geq 0 \}
\]
that achieve the same value of $J$ and call it the level set at $J = J_0$. Then, a control vector $\mathbf{x}$ is said to be time optimal if it is in the highest level set $\mathcal{L}(J = J_{\text{max}})$, where $J_{\text{max}}$ is the maximal value of $J$, and its corresponding $T$ is minimal among all control vectors in $\mathcal{L}(J = J_{\text{max}})$; a control vector $\mathbf{x}$ is nearly time optimal if it belongs to a slightly lower level set $\mathcal{L}(J = J_H)$, where $J_H \ll J_{\text{max}}$, and its corresponding $T$ is minimal among all control vectors in $\mathcal{L}(J = J_H)$.

III. THE ALGORITHM FOR SEARCHING NEAR-TIME-OPTIMAL CONTROLS

In this section, we present the two-stage iterative algorithm for seeking a near-time-optimal control vector.

A. Basic concepts

Since the objective function $J$ depends on both $T$ and $\mathbf{u}$, one could follow the gradient $dJ/d\mathbf{x}$ to maximize $J$, during which $T$ and $\mathbf{u}$ are simultaneously updated. However, there is no guarantee that $T$ will always decrease in this process (i.e., $\partial J/\partial T < 0$); even if $T$ can be decreased, the search may be trapped at a non-time-optimal solution (i.e., especially in the regime when the near-minimal value of $T$ becomes a limiting resource), or the search may cease at a negative value of $T$ that is physically unacceptable.

To reconcile the desires of increasing $J$ and decreasing $T$, one can start from some sufficiently large $T$ and look for the solution that maximizes $J$, then design an algorithm to only decrease $T$ without lowering $J$. Due to the vanishing of the gradient vector, full realization of this procedure requires that the Hessian of $J$ be calculated at the achieved maximum value, which can be computationally expensive. However, the search for a near-time-optimal control in a slightly lower level set can be much more efficient because one can make use of the gradient vector that is still dominant. In this regard, we propose the following two-stage iteration strategy:

Stage 1. Fix $T$ and use the gradient vector $dJ/d\mathbf{u}$ to increase $J$ with an arbitrary gradient-based algorithm (possibly with constraints on the control field), until $J$ reaches a prescribed high value $J = J_H$.

Stage 2. Choose a direction $\Delta \mathbf{x}$ to update $\mathbf{x}$, along which $T$ decreases most rapidly in the level set $\mathcal{L}(J = J_H)$. Considering the inevitable numerical errors in executing the algorithm, we return to stage 1 to bring the search back to the level set $\mathcal{L}(J = J_H)$ when $J$ drops to a prescribed lower bound $J = J_1$.

As schematically illustrated in Fig. 1, we keep shortening the time duration $T$ by iteratively repeating the above two-stage processes. The procedure continues until the objective $J$ can no longer be maintained at $J = J_H$. Then the resulting $T$ is the minimal time required for reaching the level set $\mathcal{L}(J = J_H)$, and the corresponding $\mathbf{u}$ is called a near-time-optimal control vector. If one wishes to find the control in a very high level set, the Hessian, whose null space can be used to refine $\mathbf{x}$ further, may need to be taken into account; the present paper does not take this last step.

Note that since the gradient-based algorithm is local, there is no guarantee that searches from any initial guesses will lead to the desired near-time-optimal solutions (i.e., shrinking $T$...
may significantly constrain the search when \( x \) enters a regime where the action of the drift Hamiltonian is encroached. Nevertheless, as shown later in examples, near-time-optimal solutions can usually be found. Even if a less than best \( T_{\text{min}} \) is obtained, the resulting control solution is still useful in practice since it can effectively reduce the time for control.

### B. Optimal choice of \( \Delta x \)

Suppose that the current value of the control vector is \( x = c \in \mathcal{L}(J = J_H) \) and we look for the direction along which \( T \) decreases most rapidly and \( J = J_H \) remains invariant. Let \( \nabla J(c) \) be the gradient vector of \( J \) with respect to \( x \) at \( c \), whose explicit expressions are given in the Appendix. To keep the search in the level set \( \mathcal{L}(J = J_H) \), the algorithm should iterate along a direction \( \Delta x \) that is orthogonal to \( \nabla J(c) \). The first-order Taylor expansion in a small neighborhood of \( c \) gives

\[
J(c + \Delta x) \approx J(c) + [\nabla J(c)]^T \Delta x.
\]

Let \( S \) be the \( mM \)-dimensional hyperplane that is orthogonal to \( \nabla J(c) \) and denote by

\[
P = I - [\nabla J(c)][\nabla J(c)]^T / ||\nabla J(c)||^2
\]

the projector onto \( S \), where \( I \) denotes the identity operator. Then, \( \Delta x \) can be written as

\[
\Delta x = Pk,
\]

where \( k \) is an arbitrary vector to be determined below.

Since the first entry of \( \Delta x \) represents the decreasing speed of \( T \), \( \Delta x^\ast \) should be chosen such that

\[
e^T \Delta x^\ast = \min(e^T \Delta x) = \min([Pe] k),
\]

where \( e = [1, 0, \ldots, 0]^T \). Thus,

\[
\Delta x^\ast \propto P(-Pe) = -Pe,
\]

which means \( k \) must be antiparallel with \( Pe \). Substituting (6) into (9), we obtain the optimal direction

\[
\Delta x^\ast \propto -e + \frac{e^T \nabla J(c)}{||\nabla J(c)||^2} \cdot \nabla J(c).
\]

As indicated in Fig. 1, expression (10) reveals that the optimal direction \( \Delta x^\ast \) is a weighted linear combination of \(-e\) and the gradient \( \nabla J(c) \). The component \(-e\) is responsible for reducing the time duration \( T \) toward a near-time-optimal control vector, but merely along this direction the iteration will be steered off the level set \( \mathcal{L}(J = J_H) \). The combination with the component \( \nabla J(c) \) compensates to keep \( J = J_H \) in the level set; the compensation will be more effective if the Hessian can be taken into account (we will not consider it in this paper). Thus, the search following this direction shortens \( T \) and maintains \( J \) at the same time and hence is expected to be more efficient than merely reducing \( T \) or climbing along the gradient direction.

### IV. NUMERICAL IMPLEMENTATION

In this section, we apply the algorithm to gate and state operations in nuclear magnetic resonance (NMR) systems that are often used for testing quantum control protocols [42,43].

<table>
<thead>
<tr>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( C_3 )</th>
<th>( C_4 )</th>
<th>( C_5 )</th>
<th>( C_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>17662</td>
<td>53.9</td>
<td>5382.4</td>
<td>3.8</td>
<td>33.96</td>
<td>4006.7</td>
</tr>
<tr>
<td>2.47</td>
<td>0</td>
<td>33.96</td>
<td>2435.8</td>
<td>0</td>
<td>3.03</td>
</tr>
<tr>
<td>34.73</td>
<td>2216.6</td>
<td>0</td>
<td>3.03</td>
<td>0</td>
<td>3.03</td>
</tr>
<tr>
<td>2105.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**A. System parameters**

We pick several examples from the six homonuclear carbon spins in the D-norleucine molecule, which are encoded as qubits for NMR quantum computation. The system’s Hamiltonian reads

\[ H = H_0 + u_1(t)H_x + u_2(t)H_y, \]

where \( u_1(t) \) and \( u_2(t) \) are the controls implemented by radio-frequency magnetic fields, which satisfy \( u_1^2(t) + u_2^2(t) \leq 1 \). Consider \( m \) spins of the six and ignore (or decouple) their couplings to the remaining spins. The drift and control Hamiltonians are as follows:

\[ H_0 = \hbar \sum_{k=1}^{m} \delta_k \omega_0 S_k^z + 2\pi \hbar \sum_{1 \leq j < k \leq m} J_{jk} (S_j^x S_k^x + S_j^y S_k^y + S_j^z S_k^z), \]

(12)

\[ H_{x,y} = -\hbar \Omega \sum_{k=1}^{m} (1 - \delta_k) S_k^{x,y}, \]

(13)

where

\[ S_k^{x,y,z} = \frac{1}{2} I_2^{(m-k)} \otimes \sigma_{x,y,z} \otimes I_2^{(m-k)} \]

are spin operators acting on the \( k \)th spin, \( \otimes \) represents the Kronecker product, and \( \sigma_{x,y,z} \) represent the standard Pauli operators. The chemical shift \( \delta_k \omega_0 \) of each spin and \( J \)-coupling constants \( J_{jk} \) between them are listed as the diagonal and off-diagonal elements in Table I. The parameter \( \Omega \) is the bound on the amplitude of control fields.

### B. Two-qubit and multiple-qubit gate optimization

First, we consider \( m = 2 \) for a two-spin example with \( C_1 \) and \( C_2 \) and seek the time-optimal control for implementing a selective \( \frac{\pi}{2} \) rotation on spin \( C_1 \), which corresponds to maximization of \( J_1 \) with \( W = \exp(-i \frac{\pi}{2} S_1^z) \otimes I_2 \).

We begin with \( T_{\text{initial}} = 200 \mu \text{s} \) and divide it into \( M = 250 \) steps. Starting from a random field as the initial guess, we obtain the optimization result [the red (dark gray) line] shown in Fig. 2(a) with error thresholds \( (1 - J_H) = 1 \times 10^{-4}, (1 - J_L) = 1.1 \times 10^{-4} \). The search ceases at \( T_{\text{min}} = 154.9 \mu \text{s} \) with error \( (1 - J_1) = 1.07 \times 10^{-4} \) after about 8000 iterations.

As displayed in Fig. 2(a), \( T \) decreases quickly in the first 500 iterations, during which it is relatively easy to stay
in the level set \( \mathcal{L}(J = J_H) \); then more and more iterations are required for corrections. To verify whether the \( T_{\text{min}} \) we obtained is the genuine minimum time duration, we test three other initial guesses (two random fields and a null field), from which all of the searches converge to the same \( T_{\text{min}} \) but at different convergence rates. In particular, the case of the null field takes more than twice the number of iterations of the first case [red (dark gray)] to reach \( T_{\text{min}} \).

To test how close our method can find a minimum time-optimal control, we raised the control amplitude \( \Omega \) from 30 to 300 kHz. As analyzed in [29], the theoretical minimal time \( T_{\text{min}} \) is supposed to approach \( T_{\text{geodesic}} = 20.4 \mu s \) calculated by Eq. (14) in [29] when \( \Omega \) is sufficiently large. We apply the algorithm with different control bounds \( \Omega \) and depict corresponding minimal times in Fig. 2(b), which shows that \( T_{\text{min}} \) does approach the theoretical limit \( T_{\text{geodesic}} = 20.4 \mu s \) when \( \Omega \) goes to infinity (e.g., \( T_{\text{min}} = 25.1 \mu s \) when \( \Omega = 300 \) kHz). This demonstrates that our algorithm is capable of finding the true time-optimal control fields in this case.

To explore more complex structures, we simulate a four-spin example with spins \( C_1, C_2, C_3 \) and \( C_4 \) and seek time-optimal control for implementing a selective \( \pi \) rotation \( W = \exp(-i S_z) \otimes I_8 \) on spin \( C_1 \). We begin with \( T_{\text{initial}} = 1000 \mu s \) and divide it into \( M = 1000 \) steps. Starting from random fields as initial guesses, we obtain the optimization results (the blue and red lines) shown in Fig. 3 with error thresholds \( 1 - J_H = 1 \times 10^{-3}, (1 - J_L) = 1.1 \times 10^{-3} \). The search ceases at \( T_{\text{min}} = 474.1 \mu s \) with error \( (1 - J_1) = 1.04 \times 10^{-3} \) after about 10 000 iterations. However, as shown in Fig. 3, the search appears less efficient when it starts from the other two randomly chosen initial guesses (the pink lines), which reveals the increasing complexity in the control Pareto behavior of large-number qubit systems. This shows that not only the dimensionality but also the potential traps for the search will increase the search effort as \( T \) is limited at a limited control resource. In such cases, one needs to more carefully choose the initial guess for \( T \) and \( u \) or develop better strategies for scanning over possible initial guesses [33].

C. Observable control optimization and comparison to other algorithms

In this case, we apply the algorithm to the objective function \( J_2 \) to maximize the expectation value of the observable \( O \) and compare our algorithm with another time-optimal control procedure.

We simulate a two-spin example \((C_1 \text{ and } C_2)\) with the following initial state and observable:

\[
\rho_0 = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}, \quad O = \begin{pmatrix}
\frac{1+i}{2} & 0 & \frac{\sqrt{2}}{2} \\
0 & 0 & 0 \\
-\frac{\sqrt{2}}{2} & 0 & \frac{1+i}{2} \\
0 & 0 & 0
\end{pmatrix}.
\]

(14)

It can be verified that the maximal value of \( J_2 \) is \( J_{\text{max}} = 1 \).

We begin with \( T_{\text{initial}} = 60 \mu s \) and divide it into \( M = 100 \) steps. Starting from a random field as the initial guess, we
obtain the optimization result (the red line) shown in Fig. 4(a) with error thresholds $(1 - J_H) = 1 \times 10^{-6}, (1 - J_L) = 1.1 \times 10^{-6}$. The search ceases at $T_{\text{init}} = 58.18 \mu s$ with error $(1 - J_L) = 1.00 \times 10^{-6}$ after about 20 000 iterations. Figure 4(b) shows the corresponding time-optimal control functions. The green line represents the constraint of the control amplitude, which is shown at the highest level at all times. It could be taken as the feature of time-optimal solutions.

We then compare the proposed LS algorithm with an existing algorithm in the literature [18,34], which decreases $T$ step by step by a fixed size $\Delta T$. Figure 4(a) shows the results with step size $\Delta T = 0.01 \mu s$ and $\Delta T = 0.1 \mu s$, respectively. For the bigger step size $\Delta T = 0.1 \mu s$, $T$ decreases faster than in our algorithm in the first 5000 iterations but slower in the last 25 000 iterations and finally ceases at $T = 58.2 \mu s$ after 25 000 iterations (light gray, lower line); for the smaller step size $\Delta T = 0.01 \mu s$, $T$ decreases so slow that no satisfying value can be obtained within 50 000 iterations (light blue, upper line). Thus, the LS algorithm achieves overall good performance with more rapid convergence and higher final precision under the same computational burden.

**FIG. 4.** (Color online) (a) The searches for near-time-optimal controls for the observable control over spins $C_1$ and $C_2$ with error thresholds $[1 \times 10^{-6}, 1.1 \times 10^{-6}]$. $T$ decreases from $T_{\text{init}} = 60 \mu s$ to $T_{\text{min}} = 58.18 \mu s$ after 20 000 iterations (our algorithm; red), $T_{\text{min}} = 58.20 \mu s$ after 25 000 iterations ($\Delta T = 0.1 \mu s$; light gray), and $T_{\text{min}} = 58.35 \mu s$ after 30 000 iterations ($\Delta T = 0.01 \mu s$; light blue). (b) The optimal control functions $u_1(t)$ (red, right) and $u_2(t)$ (blue, left) in the two-qubit observable control example. Each vertical line represents a value of control variable $u_k(t_k)$, and the green line shows the numerical result of $u_1^2(t_j) + u_2^2(t_j)$ at different $t_j$.

To summarize, we present a level-set-based iterative algorithm for searching near-time-optimal controls of quantum systems. Operating near the prescribed level set, the algorithm seeks high-quality time-optimal solutions by continuously updating the control functions and intermittently decreasing the time duration, in which the searches are able to follow the optimal direction, which takes care of both the fidelity $J$ and the time duration $T$. Numerical examples show that the algorithm can quickly find near-time-optimal solutions for both gate and observable control problems. The central idea of searching in high level sets can be generalized with other local optimization algorithms with fixed $T$, and the control profiles can be further smoothed and optimized with advanced algorithms such as the hybrid procedure proposed in [33], so the quality and robustness of the time-optimal control can be improved.

In principle, the proposed algorithm can also be applied to open quantum systems with nonunitary dynamics. However, it is not always easy to determine a proper high level set that is reachable by admissible controls. Moreover, due to the complexity of the open system dynamics, we expect that the search will encounter more traps than in the case of unitary dynamics. Nevertheless, since unitary dynamics is often a good approximation of open system dynamics within a short time interval, we can reasonably consider the unitary case under most circumstances. Even if the unitary approximation is not adequate, we can still use the obtained time-optimal control as an initial guess for optimization with nonunitary dynamics.

**ACKNOWLEDGMENTS**

This work is supported by the National Natural Science Foundation of China (Grants No. 61134008, No. 61374091, and No. 91221205) and the Army Research Office (Grant No. W911NF-11-1-2068). We thank Prof. X. Peng from the University of Science and Technology of China for useful discussions and the NMR data for simulations, and we thank W. Wang for work on the coding.

**APPENDIX: THE GRADIENT FORMULAS**

For convenience, we denote

$$U_j = V_j V_{j-1} \cdots V_2 V_1, \quad j = 1, 2, \ldots, M,$$

where

$$V_j = \exp \left[ -i \left( H_0 + \sum_{k=1}^M u_k \, H_k \right) \frac{T}{M} \right].$$

When $\frac{T}{M}$ is sufficiently small, we have [18]

$$\frac{\partial V_j}{\partial u_k} \approx -\frac{i}{M} H_k V_j,$$

$$\frac{\partial V_j}{\partial T} = -\frac{i}{M} \left( H_0 + \sum_{k=1}^M u_k \, H_k \right) V_j.$$
Using the above results, we obtain the total derivative of $U(T) = U_M$ with respect to $x$,

$$\frac{\partial U(T)}{\partial u_{kj}} = -\frac{i}{M}U(T)U^\dagger_k H_k U_j,$$

$$\frac{\partial U(T)}{\partial T} = -\frac{i}{M}\sum_{j=1}^{M} U(T)U^\dagger_j \left( H_0 + \sum_{k=1}^{m} u_{kj} H_k \right) U_j.$$

The above formulas can be directly applied to objective functions (2) and (3) to obtain the following gradient formula:

$$\frac{\partial J_i}{\partial u_{kj}} = \frac{T}{M} \text{Im}(\text{Tr}(D_i U_j^\dagger H_j U_j)),$$

$$\frac{\partial J_i}{\partial T} = \frac{1}{M} \sum_{j=1}^{M} \text{Im}\left\{ \text{Tr}\left[ D_i U_j^\dagger \left( H_0 + \sum_{k=1}^{m} u_{kj} H_k \right) U_j \right] \right\},$$

where $\text{Im}(\cdot)$ denotes the imaginary part, $i = 1, 2, k = 2, \ldots, m$, and $j = 1, \ldots, M$, with

$$D_1 = W^\dagger U_M,$$

$$D_2 = [\rho_0, U_M^\dagger O U_M].$$


[44] University of Science and Technology of China (unpublished).