Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 360 (2023) 13972–13993

www.elsevier.com/locate/jfranklin

Two-step robust control design of quantum gates via differential evolution☆

Shouliang Hu[a], Hailan Ma[b], Daoyi Dong[b], Chunlin Chen[a,*]

[a] *Department of Control and Systems Engineering, Nanjing University, Nanjing, 210093, China*
[b] *School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia*

Received 30 March 2022; received in revised form 20 May 2022; accepted 10 June 2022
Available online 20 June 2022

Abstract

Designing highly accurate and robust controls for quantum unitary operations is vital for practical quantum computation. In this paper, we demonstrate that the robustness of quantum gate controls can be enhanced by optimizing the sampling-based infidelity variance, which is formulated as a multi-objective optimization task to achieve high robustness while maintaining high gate fidelity. A two-step approach that first optimizes the average fidelity and then turns to the infidelity variance is proposed, where two modified differential evolution (DE) algorithms, i.e., mixed-guided-strategy DE (MGSDE) and multi-objective mixed-strategy DE (MOMSDE), are designed to search fields for the two steps of robust quantum gate control, respectively. Both MGSDE and MOMSDE adopt the mixed strategy, and in particular, MGSDE adopts a guided mutation scheme to accelerate the convergence, and MOMSDE uses an optimal buffer to explore the Pareto front. Numerical results demonstrate that the proposed approach enhances the control robustness and provides an efficient in-situ learning paradigm to tackle the disturbance problem for the control design of quantum gates.

☆ This work was supported in part by the National Natural Science Foundation of China (Nos.62073160 and 61828303) and the Australian Research Council's Discovery Projects funding scheme under project DP190101566.
* Corresponding author.
E-mail address: clchen@nju.edu.cn (C. Chen).

https://doi.org/10.1016/j.jfranklin.2022.06.014
0016-0032/

1. Introduction

Quantum information technology has attracted much attention in the past decades, as quantum computers have shown promise for solving specific computational tasks such as finding the prime factors [1,2]. One property distinguishing a quantum computer from its counterparts is that the quantum computer uses the quantum analogs of bits and logic gates, referred to as quantum gates, to accomplish information processing [3]. The implementation of quantum computers has specific criteria, including a "universal' set of quantum gates, which may consist of some single-qubit operations and a two-qubit CNOT gate, as any computation process can be decomposed into a sequence of these basic gate operations [2]. Hence, the control design of quantum gates lies at the heart of practical quantum information technology [4].

An important issue when designing quantum controls is the extent that noises and parameter fluctuations hinder control performance [5]. In laboratory devices, fluctuations like charge noise in the coupling term [6], random signal jitters, and channel latencies in the control electronics [7] significantly increase the operation error of quantum control tasks. Therefore, designing robust control strategies compensating for the influence of disturbances is vital for practical quantum devices.

Recently, sampling-based optimization has been systematically applied to the robust control design of quantum systems and quantum operations [7–22]. The core idea thereof is to optimize the fitness regarding neighborhood values of uncertainties [12]. Most of the work sets a single control objective, which improves the robustness of control $\boldsymbol{u}$ by maximizing the average fidelity or optimizing the worst-case fidelity. For example, researchers have realized robust state transition control of inhomogeneous quantum ensembles of spins, as well as Λ type atomic systems via optimizing the average performance [12]. Robust quantum transformations in a superconducting circuit and a spin chain system are achieved by optimizing the mean value [21]. Using a controlled one-qubit system, a higher degree of robustness can be obtained by optimizing the worst-case gate fidelity [9].

From the point of view of optimization, robustness measures the sensitivity of fitness variation to parameter fluctuations. The landscape of a two-variable function is shown in Fig. 1. If the optimization objective is to achieve high fitness, solution $\boldsymbol{a}$ is optimal. However, the solution $\boldsymbol{a}$ is non-robust as the fitness varies significantly with the perturbation. From a geometric perspective, solution $\boldsymbol{b}$ possesses a shallow curvature than $\boldsymbol{a}$, therefore giving better robustness as it dampens the effect of variable variations [5,23]. Various methods that consider fitness deviation due to high-order control perturbations have been developed to flatten the robustness control landscape for quantum control tasks [24–26]. For example, for the optimal control of a single-phase space trajectory, a gradient-based search algorithm based on D-MORPH is implemented to optimize the fitness perturbation due to a control error up to second-order [24]. Global robust optimal control strategies for control of two-level quantum systems are obtained using the Pontryagin maximum principle and could make the NOT gate robust against uncertainty to the second-order [25]. A risk-sensitive method taking advantage of optimizing the worst-case and average infidelity achieves high control robustness, as it achieves control over the variance of the error [26]. In this paper, we try to optimize the sampling-based fitness variance, which helps flatten the robustness control landscape and is easy to calculate.

Another challenging task for robust quantum control lies in the optimization algorithm, which significantly influences the performance. Based on the gradient-based method of GRAPE [27], several robust approaches have been proposed for robust quantum control tasks

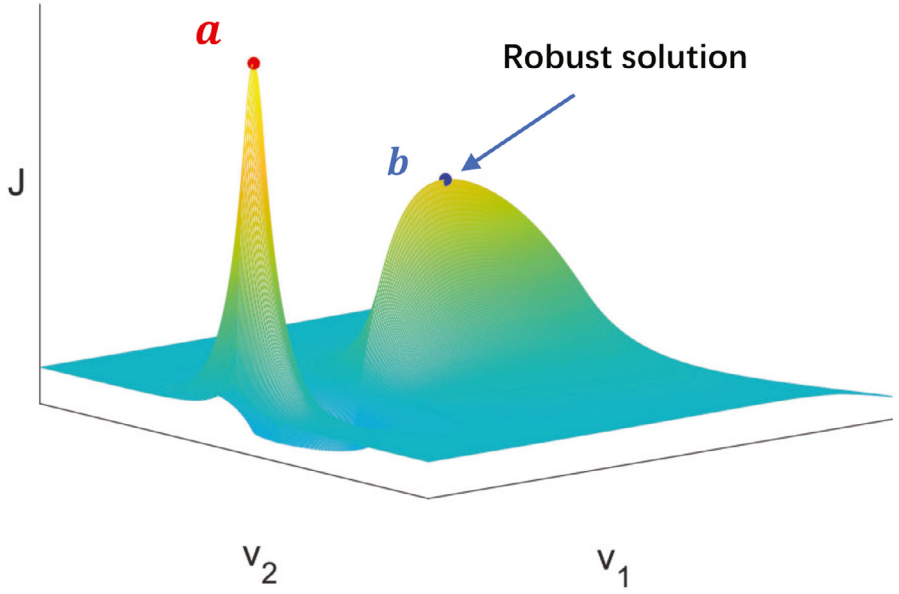


Fig. 1. Landscape regarding robustness of the function $J[v_1, v_2] = 2sin(10v_1e^{-0.5v_1})e^{-0.25v_1}e^{-(5-v_1)|v_2-2|}$.

[7,26,28,29]. For example, b-GRAPE utilizes the trick of batch-based optimization and randomly generates samples for each optimization cycle, and shows that it can effectively improve control robustness against Hamiltonian uncertainties and clock noises [7,28]. An adversarial gradient-based learning algorithm (a-GRAPE) can learn highly robust controls by generating adversarial samples through the search for Nash equilibria [29]. Such gradient-based algorithms converge fast and could provide single-shot controls that achieve high robustness.

However, robust quantum control problems that have complex optimization landscapes may trap gradient-based methods to local optima [30]. For these situations, this paper explores the use of stochastic algorithms, especially evolutionary algorithms. In the 1990s, Rabitz first applied the genetic algorithm (GA) to control the laser excitation of molecules with experimental feedback [31]. After that, GA has been applied to various quantum control problems, such as searching control pulses for state preparation and quantum gate operations in NMR [32] and controlling the ionization pathway of a Rydberg electron [33]. In 1997, differential evolution (DE) was proposed and had been widely used in a wide range of optimization tasks because of its simplicity and effectiveness [34]. DE directly encodes the parameters to be optimized into a vector of real numbers, which makes DE have a greater degree of computational complexity for combinatorial problems than GA [35]. In addition, DE can explore better solutions for continuous optimization and has been successfully applied to various quantum control problems [16,22,36,37]. For example, a subspace-selective self-adaptive differential evolution (SUSSADE) algorithm has been applied to designing three-qubit Toffoli, Controlled-Not-Not, and Fredkin gates [36,37]. An equally-mixed strategies DE (EMSDE) algorithm has successfully tackled the state-to-state transition task of an open quantum system [16]. Yang and co-workers proposed an improved DE algorithm to solve the quantum state and gate preparation problems [38]. A comparative study shows that GRAPE

is sensitive to control imperfections and measurement errors, but DE performs much better [39].

In this paper, we adopt multi-objective analysis and apply it to the robust control design of quantum gates. The control precision and robustness are measured by the mean fidelity and the infidelity variance in response to sampled disturbances. Since quantum gate operation requires high precision and the optimization landscape with uncertainties is highly complicated, to speed up the search of optimal control variables, we propose an approach that divides the optimization into two steps, including (i) fidelity optimization and (ii) robustness optimization. The fidelity optimization step aims at improving the average gate fidelity, and we develop a mixed-guided-strategy DE (MGSDE) algorithm using a modified guiding scheme [40] and several mixed mutation strategies. The robustness optimization step inherits the population obtained in the first step of fidelity optimization and then performs the multi-objective optimization, where we propose a multi-objective mix-strategy DE (MOMSDE) algorithm that adopts an optimal buffer to explore the Pareto solutions. The obtained Pareto fronts denote the degradation of robustness (e.g., infidelity variance) regarding the rising of optimal performance (e.g., average fidelity). Solutions in the obtained Pareto fronts can meet different requirements of optimality and robustness. Compared to the traditional approach that only optimizes the mean performance, our method can remarkably reduce the fidelity dispersion while maintaining high precision.

The rest of the paper is organized as follows. Section 2 gives the problem formulation and briefly introduces multi-objective optimization and differential evolution. Section 3 presents the framework of the two-step optimization approach and the detailed implementation with two modified differential evolution algorithms. Section 4 presents the numerical results of robust control of three one-qubit gates and the CNOT gate. The conclusion is given in Section 5.

2. Preliminaries

2.1. Problem formulation

Considering an N-dimensional closed quantum system, the state of the system can be represented with $|\psi\rangle$ (Dirac notation), where $|\psi\rangle \in \mathbb{C}^N$ is a complex vector. The state transformation can be described by

$$|\psi(t)\rangle = U(t)\,|\psi_0\rangle\,, \tag{1}$$

where $U(\cdot) \in \mathbb{C}^{N\cdot N}$ represents the unitary operator (the quantum logic gate). The propagator $U(t)$ starts from $U(0) = \mathbb{I}_N$ and is governed by the Schr-dinger equation (setting the Plank constant $\hbar = 1$):

$$\frac{d}{dt}U(t) = -iH[t;\boldsymbol{u},\boldsymbol{\epsilon}]U(t), \tag{2}$$

where $\mathbb{I}_N$ is the identity matrix. Considering the existence of parameter uncertainties $\boldsymbol{\epsilon} = [\omega, \theta]$, the Hamiltonian of the system is given as

$$H[t;\boldsymbol{u},\boldsymbol{\epsilon}] = f_1(\omega)H_0 + f_2(\theta)H_c[t;\boldsymbol{u}], \tag{3}$$

where the functions $f_1(\omega)$ and $f_2(\theta)$ characterize the possible parameter fluctuations in the free Hamiltonian H_0 and the time-varying control Hamiltonian $H_c[t;\boldsymbol{u}]$, respectively. For simplicity, this paper assumes that $f_1(\omega) = 1 + \omega$, $f_2(\theta) = 1 + \theta$ and θ, ω are time-independent

uncertainties, which are distributed uniformly, i.e., $\omega, \theta \in [-E, E]$. Here, $\mathbf{u}$ is the design variable to be optimized, which represents the laboratory generated control field interacting with the quantum system. In experimental implementations, the piece-wise constant approach would equally divide the transfer time T into K intervals and keeps the control amplitude constant during each interval, i.e. $\boldsymbol{u} = [u_1, u_2, \cdots, u_K]$ [27,41]. The unitary transformation U_j during the jth interval satisfies

$$U_j = \exp\left(-i \int_{(j-1)\triangle T}^{j\triangle T} H[t; u_j, \boldsymbol{\epsilon}]dt\right). \tag{4}$$

The gate $U(T)$ at the final time T is given as $U(T) = U_k \cdots U_2 U_1$. A common way to assess the performance under the control $\boldsymbol{u}$ and the uncertainty $\boldsymbol{\epsilon}$ is to compute the gate fidelity, which is defined as

$$F(\boldsymbol{u}, \boldsymbol{\epsilon}) = \frac{1}{N^2}\left|tr\left\{U_f^{\dagger} U(T)\right\}\right|. \tag{5}$$

The general goal of quantum gate control is to manipulate the system dynamics for achieving unitary operations with high fidelities. However, parameter uncertainties severely hider the control performance. The robust control aims at achieving high fidelities for all possible variations so as to make the control performance insensitive to parameter fluctuations.

To achieve high control precision, we may optimize the average fidelity $\mathbb{E}(\boldsymbol{u})$:

$$\mathbb{E}(\boldsymbol{u}) = \int_{\boldsymbol{\epsilon}} F(\boldsymbol{u}, \boldsymbol{\epsilon}) P(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}, \tag{6}$$

where $P(\boldsymbol{\epsilon})$ represents the probability distribution of $\boldsymbol{\epsilon}$. Better control precision is given with the rising of $\mathbb{E}(\boldsymbol{u})$.

To achieve high control robustness, a proper measure of performance dispersion is vital. This paper measures the control error using the gate infidelity:

$$L(\boldsymbol{u}, \boldsymbol{\epsilon}) = \log_{10}(1 - F(\boldsymbol{u}, \boldsymbol{\epsilon})). \tag{7}$$

Then the infidelity variance according to the resulting errors is used to evaluate the sensitivity (or robustness):

$$\sigma^2(\boldsymbol{u}) = \int_{\boldsymbol{\epsilon}} |L(\boldsymbol{u}, \boldsymbol{\epsilon}) - \mu(\boldsymbol{u}, \boldsymbol{\epsilon})|^2 P(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}. \tag{8}$$

where $\mu(\boldsymbol{u}, \boldsymbol{\epsilon}) = \int_{\boldsymbol{\epsilon}} L(\boldsymbol{u}, \boldsymbol{\epsilon}) P(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}$ is the mean value of infidelity. The infidelity L involves non-linearity, which is more sensitive to variations of $\boldsymbol{\epsilon}$ than the fidelity F. Hence, the variance of infidelity gives a better measurement of performance dispersion. Better robustness is given with the declining of $\sigma^2(\boldsymbol{u})$.

Thus the task can be formulated as a multi-objective problem:

$$\begin{aligned}
&\max_{\boldsymbol{u}} \mathbb{E}(\boldsymbol{u}) \\
&\min_{\boldsymbol{u}} \sigma^2(\boldsymbol{u}) \\
&s.t.\ \ \frac{d}{dt} U(t) = -iH[t; \boldsymbol{u}, \boldsymbol{\epsilon}] U(t), \\
&U(0) = \mathbb{I}_N, \\
&H[t; \boldsymbol{u}, \boldsymbol{\epsilon}] = f_1(\omega) H_0 + f_2(\theta) H_c[t; \boldsymbol{u}], \\
&t \in [0, T], \quad \boldsymbol{\epsilon} = [\omega, \theta] \in \triangle.
\end{aligned} \tag{9}$$

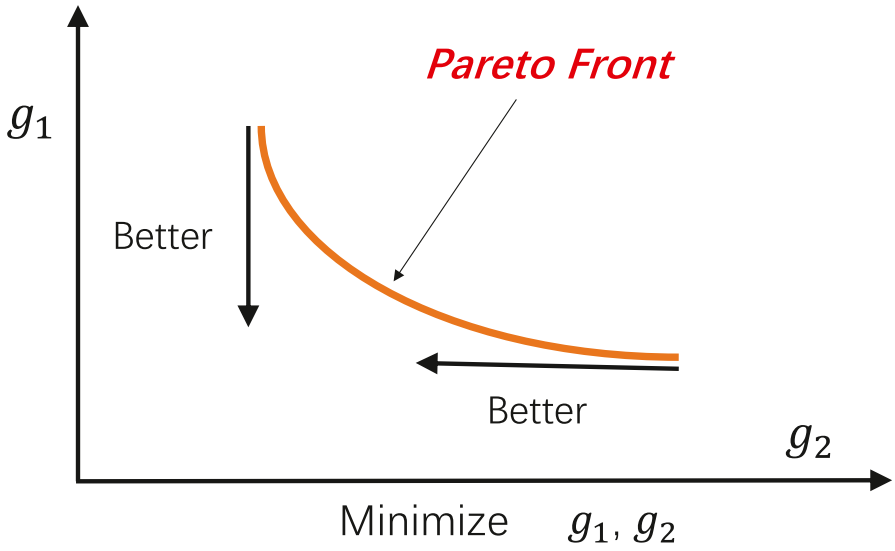


Fig. 2. Description of Pareto front.

The robust design aims at finding a solution $\boldsymbol{u}^*$ that makes a good balance between robustness and precision.

2.2. Multi-objective optimization

The general multi-objective optimization (MO) task aims at optimizing N objectives simultaneously as [42]

$$\begin{aligned} &\text{Minimize} \quad \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \cdots, g_N(\boldsymbol{x})] \\ &\text{Subject to} \quad h_i(\boldsymbol{x}) \leqslant 0, i = 1, 2, \cdots, M \\ &\qquad\qquad \boldsymbol{x} \in \Omega \end{aligned} \tag{10}$$

where $\boldsymbol{y}$ is the objective vector, $\boldsymbol{x}$ consists of the decision variables that belong to a parameter space Ω and h_i represents the constraints. In an MO problem, a solution $\boldsymbol{x}_1$ dominates $\boldsymbol{x}_2$ means that $\boldsymbol{g}(\boldsymbol{x}_1)$ is better than $\boldsymbol{g}(\boldsymbol{x}_2)$ in all objectives, as

$$g_i(\boldsymbol{x}_1) \leqslant g_i(\boldsymbol{x}_2), \quad \text{for all} \quad i \in \{1, 2, \cdots, \mathrm{N}\}. \tag{11}$$

For an MO problem, the global optimum does not exist if the individual objective conflicts with each other. In such cases, a set of non-dominated solutions can be obtained. For a solution $\boldsymbol{u}$, if there is no other solution that dominates it in the whole parameter space Ω, $\boldsymbol{u}$ satisfies Pareto optimal property. The set of all Pareto optimal solutions is named as the Pareto optimal front set that pictures the trade-offs between different objectives (as shown in Fig. 2) [42]. MO optimization aims at finding a set of non-dominated solutions closed to the actual Pareto front, which provides various options to meet the requirement of the problem.

2.3. Differential evolution

Differential evolution (DE) is a population-based global optimization algorithm that encodes the parameters to be optimized into vectors of real numbers, referred to as individual trail solutions. The population consisits of NP K-dimensional vectors, i.e., $X_{i,G} = \{x_{i,G}^1, \cdots, x_{i,G}^K\}$, $i = 1, \cdots, NP$ [43]. At the beginning, the individuals are initialized randomly, as the jth component of the ith individual is set as $x_i^j = x_{\max} + \text{rand}(0, 1) \cdot (x_{\max} - x_{\min})$, where $x_{\max}$, $x_{\min}$ are hyper-parameters that bounds the searching space and $\text{rand}(0, 1)$ is a uniformly distributed random number. Then, the population is evolved through three operations, including *Mutation*, *Crossover* and *Selection*.

(a) *Mutation*. The *Mutation* operates on several randomly chosen vectors and follows a specific mutation strategy, such as 'DE/rand/1', to generate the difference vector $V_{i,G}$

$$V_{i,G} = X_{r_1,G} + \text{F} \cdot (X_{r_2,G} - X_{r_3,G}), \tag{12}$$

where F is a user-specified positive real number named as difference factor and $X_{r_1,G}$, $X_{r_2,G}$, $X_{r_3,G}$ are individual vectors that are randomly selected from the current population.

(b) *Crossover*. Inspired by the chromosome crossover phenomenon in nature, DE applies the binomial crossover operation on the current trail solutions $X_{i,G}$ and the mutant vector $V_{i,G}$ to generate a new trail vector $U_{i,G} = \{u_{i,G}^1, \cdots, u_{i,G}^K\}$ following

$$u_{i,G}^j = \begin{cases} V_{i,G}^j, & \text{if rand}(j) \le \text{CR or } j = j_{\text{rand}}, \\ X_{i,G}^j, & \text{if rand}(j) > \text{CR or } j \ne j_{\text{rand}}, \end{cases} \quad j = 1, \cdots, K \tag{13}$$

Here, CR is the crossover rate within the range [0,1) that controls the fraction of paramter values copied from $V_{i,G}$ [43]. And j_{rand} is a integer randomly generated within the range $[1, K]$, which is used to make sure that every mutant vector suffers from at least one *Crossover* operation.

(d) *Selection*. Finally, DE computes the fitness value $f(V_{i,G})$ and $f(X_{i,G})$. Supposing that the optimization task is a maximization problem, then the vector that has higher fitness survives to the next generation,

$$X_{i,G+1} = \begin{cases} V_{i,G}, & \text{if } f(U_{i,G}) > f(X_{i,G}), \\ X_{i,G}, & \text{otherwise.} \end{cases} \tag{14}$$

DE repeats the above operations on individuals to evolve the population and outputs the best solution at the end. The optimization landscape of the robust gate control problem is highly complicated, hence implementing standard DE may be time-consuming. In this paper, we apply modified DE algorithms for the robust control design.

3. Two-step robust control design via DE

In this section, we first present the framework of the proposed two-step robust control design approach and then introduce the detailed implementations with MGSDE and MOMSDE algorithms for the two steps of optimization, respectively.

3.1. Framework of the two-step robust control design

The framework of the proposed two-step learning control approach for robust control design of quantum gates is shown in Fig. 3. As the control design focuses on performance

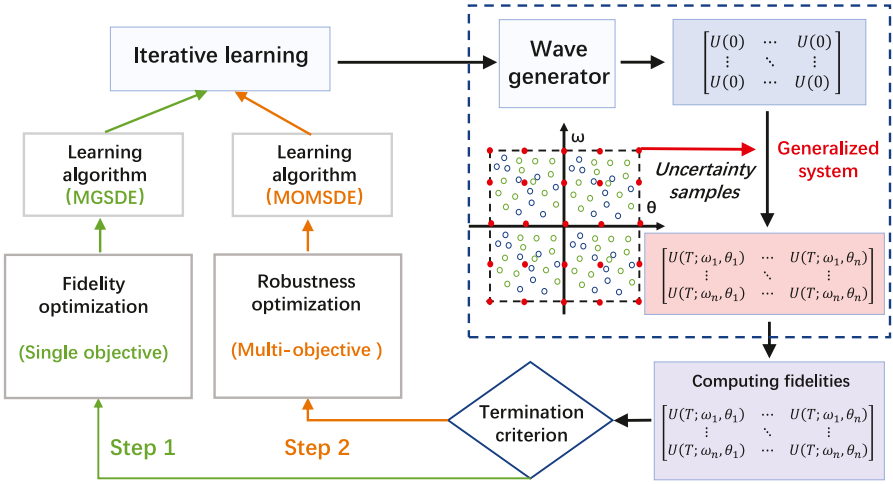


Fig. 3. Framework of the two-step robust control design for quantum gates.

and robustness, the optimization goal is twofold: (i) maintaining high gate fidelity no matter how the uncertainty varies; (ii) minimizing the sensitivity of fidelity degradation to uncertainty fluctuations.

We apply the sampling-based technique [12] to search for the optimal robust controls. To evaluate the fitness of control $\boldsymbol{u}$, we evenly sample n pairs of uncertainty parameters from the bounded disturbance interval. With respect to the samples, the expectation value defined in (6) can be approximated as the average fidelity:

$$f_1(\boldsymbol{u}) = \frac{1}{n}\sum_{k=1}^{n} F(\boldsymbol{u}, \boldsymbol{\epsilon}_k), \tag{15}$$

and the robustness measurement adopts the infidelity variance can be computed as

$$f_2(\boldsymbol{u}) = \sigma^2[L(\boldsymbol{u}, \boldsymbol{\epsilon}_k)] = \frac{1}{n-1}\sum_{k=1}^{n} |L(\boldsymbol{u}, \boldsymbol{\epsilon}_k) - \mu(\boldsymbol{u}, \boldsymbol{\epsilon}_k)|^2, \tag{16}$$

where $\mu(\boldsymbol{u}, \boldsymbol{\epsilon}_k) = \frac{1}{n}\sum_{k=1}^{n} L(\boldsymbol{u}, \boldsymbol{\epsilon}_k)$.

The control task is formulated as a multi-objective problem of (9). However, control of quantum gate requires high precision (e.g., the gate fidelity $F > 0.999$), while multi-objective evolutionary algorithms generally require a long time to achieve high performance. To reduce the computation cost, we divide the optimization into two steps of *Fidelity optimization* and *Robustness optimization*, which are described in Algorithm 1.

In the implementation, we first optimize the average performance using MGSDE algorithm shown as Algorithm 2. The criteria condition is set as achieving the maximal iteration number $G_{\max,1}$. It should be noted that numerical simulations have revealed that a DE population starting with high average fitness (i.e., high average fidelity) can explore solutions with better robustness (i.e., lower infidelity variance). Therefore, we set large iterations in the fidelity optimization step as it significantly influences the control performance. In the robustness optimization step, the MOMSDE algorithm (i.e., Algorithm 4) is implemented to optimize

Algorithm 1 Framework of two-step robust control design.

1: **Initialization**: Set the generation index $G = 0$ and randomly initialize the individuals.
2: ***Fidelity optimization***: Implement the MGSDE algorithm (i.e., Algorithm 2) to optimize the average fidelity until $G = G_{\max,1}$ and then transit the evolved individuals to the robustness optimization step.
3: ***Robustness optimization***: Reset the generation index as $G = 0$ and implement the MOMSDE algorithm (i.e., Algorithm 4) to optimize the infidelity variance and average fidelity.
4: **Output**: An optimal Pareto solution set B^* is provided by Algorithm 4 with $G_{\max,2}$ iterations.

Algorithm 2 Algorithmic description of MGSDE.

Initialization: Set the generation index $G = 0$ and randomly initialize the individuals.
Evaluate the fitness $f(X_{i,G})$ of all individuals.
repeat
  Group the individuals based on their fitness value.
  for $i = 1 : NP$ (each individual) **do**
    Mutation
    Select a guided mutation strategy from Eqs. (22)-(25) to generate a mutation vector $V_{i,G}$.
    Crossover
    Generate a trail vector $U_{i,G}$ via crossover operation.
    Selection
    Compute the fitness value of the trail vector $U_{i,G}$.
    if $f(U_{i,G}) \geq f(X_{i,G})$ **then**
      $X_{i,G} \leftarrow U_{i,G}, \quad f(X_{i,G}) \leftarrow f(U_{i,G})$.
    end if
  end for
  $G \leftarrow G + 1$.
until $G = G_{\max,1}$

the infidelity variance and average fidelity simultaneously. In the end, we can obtain solutions converging to the Pareto front with diverse degrees of robustness and optimality.

3.2. Fidelity optimization

The fidelity optimization aims at improving the gate fidelity with respect to all uncertainty values, which can be realized by optimizing the average performance as

$$\max_{\boldsymbol{u}} f(\boldsymbol{u}) = \frac{1}{n}\sum_{k=1}^{n} F(\boldsymbol{u}, \boldsymbol{\epsilon}_k).$$

$$s.t. \ \frac{d}{dt}U(t) = -iH[t; \boldsymbol{u}, \boldsymbol{\epsilon}_k]U(t),$$

$$U(0) = \mathbb{I}_N, \quad k = 1, \cdots, n,$$

$$H[t; \boldsymbol{u}, \boldsymbol{\epsilon}_k] = (1 + \omega_k)H_0 + (1 + \theta_k)H_c[t; \boldsymbol{u}],$$

$$t \in [0, T], \quad \omega_k, \theta_k \in [-E, E], \tag{17}$$

where the fitness $f(\boldsymbol{u})$ is defined as the average fidelity of the n samples. We develop a modified DE algorithm, named GMSDE, to search for optimal solutions, which is shown as Algorithm 2.

In DE, mutation operation significantly influences the performance [40] and traditional mutation strategies mainly rely on randomly chosen individuals. For example, "DE/rand/1" [34] is a fundamental and widely used mutation strategy, which is formulated as

$$V_i = X_{r_1} + \mathrm{F} \cdot (X_{r_2} - X_{r_3}), \tag{18}$$

where $X_{r_1}, X_{r_2}, X_{r_3}$ are randomly chosen from the current population. The other three mutation strategies based on the same scheme have also been used widely and are listed as follows.

"rand-to-best/2":

$$V_i = X_i + \mathrm{F} \cdot (X_{\text{best}} - X_i) + \mathrm{F} \cdot (X_{r_1} - X_{r_2}) + \mathrm{F} \cdot (X_{r_3} - X_{r_4}). \tag{19}$$

"rand/2":

$$V_i = X_{r_1} + \mathrm{F} \cdot (X_{r_2} - X_{r_3}) + \mathrm{F} \cdot (X_{r_4} - X_{r_5}). \tag{20}$$

"Guided-current-to-rand/1":

$$V_i = X_i + \mathrm{K} \cdot (X_{r_1} - X_i) + \mathrm{F} \cdot (X_{r2} - X_{r3}). \tag{21}$$

Traditional mutation strategies compute the searching direction (i.e., differential vector) based on randomly chosen vectors, which helps maintain population diversity but slows down the convergence speed [40]. Recently, to accelerate the optimization process, a "guided" mutation scheme has been proposed to give better search directions via grouping the vectors according to their fitness [40]. In such a scheme, all individuals are grouped into three groups, and a mutation trail vector is generated following

$$V_i = X_m + \mathrm{F} \cdot (X_t - X_b), \tag{22}$$

where X_m, X_t and X_b are randomly chosen from the *median*, *top* and *bottom* groups. This mutation scheme generates differential vectors utilizing the best and worst vectors to improve the convergence speed, while maintains a global exploration as it uses middle vectors as bases [40]. Inspired by this "guided" mutation scheme, we generate another three mutation strategies as follows:

"Guided-rand-to-best/2":

$$V_i = X_i + \mathrm{F} \cdot (X_{\text{best}} - X_i) + \mathrm{F} \cdot (X_{t1} - X_{b1}) + \mathrm{F} \cdot (X_{t2} - X_{b2}). \tag{23}$$

"Guided-rand/2":

$$V_i = X_{\mathrm{m}} + \mathrm{F} \cdot (X_{t1} - X_{b1}) + \mathrm{F} \cdot (X_{t2} - X_{b2}). \tag{24}$$

"Guided-current-to-rand/1":

$$V_i = X_i + \mathrm{K} \cdot (X_m - X_i) + \mathrm{F} \cdot (X_{t1} - X_{b1}). \tag{25}$$

In DE, constructing a candidate pool that consists of multiple mutation strategies has been proven superior to implementing one single mutation strategy [16,44]. Based on the "guided" mutation strategies (i.e., equations (22)-(25)), we develop a mixed-guided-strategy DE (MGSDE) algorithm as shown in Algorithm 2. In the mutation operation of MGSDE,

Table 1
Description of the mixed-strategy for MGSDE.

Mutation strategy	Condition
Guided-rand/1 (i.e., Eq. (22))	$p_i \in [0, 0.25)$
Guided-rand-to-best/2 (i.e., Eq. (23))	$p_i \in [0.25, 0.5)$
Guided-rand/2 (i.e., Eq. (24))	$p_i \in [0.5, 0.75)$
Guided-current-to-rand/1 (i.e., Eq. (25))	$p_i \in [0.75, 1]$

all candidate strategies are chosen with equal probability (i.e., 0.25). For each individual, we randomly generate a number $p_i \in [0, 1]$ and choose a mutation strategy following four cases as listed in Table 1.

Besides, inspired by the previous studies [16,43–45], we generate different mutation and crossover factors for each individual using the normal distribution, i.e., $\mathrm{F}_{i,G} \sim N(n_1, \sigma_1)$, $\mathrm{CR}_{i,G} \sim N(n_2, \sigma_2)$.

3.3. Robustness optimization

The central goal of this paper is to improve control robustness while maintaining high precision. Therefore, the robustness optimization step is formulated as a multi-objective optimization task that aims at optimizing the sampling-based infidelity variance and mean fitness simultaneously, which is formulated as

$$
\begin{aligned}
&\max_{\boldsymbol{u}} f_1(\boldsymbol{u}) = \frac{1}{n}\sum_{k=1}^{n} F(\boldsymbol{u}, \boldsymbol{\epsilon}_k).\\
&\min_{\boldsymbol{u}} f_2(\boldsymbol{u}) = \frac{1}{n-1}\sum_{k=1}^{n} |L(\boldsymbol{u}, \boldsymbol{\epsilon}_k) - \mu(\boldsymbol{u}, \boldsymbol{\epsilon}_k)|^2\\
&s.t.\ \ \frac{d}{dt}U(t) = -iH[t; \boldsymbol{u}, \boldsymbol{\epsilon}_k]U(t), \quad k = 1, \cdots, n\\
&U(0) = \mathbb{I}_N,\\
&H[t; \boldsymbol{u}, \boldsymbol{\epsilon}_k] = (1 + \omega_k)H_0 + (1 + \theta_k)H_c[t; \boldsymbol{u}],\\
&\mu(\boldsymbol{u}, \boldsymbol{\epsilon}_k) = \frac{1}{n}\sum_{k=1}^{n} L(\boldsymbol{u}, \boldsymbol{\epsilon}_k),\\
&t \in [0, T], \quad \omega_k, \theta_k \in [-E, E].
\end{aligned}
\tag{26}
$$

To solve the robustness optimization problem efficiently, we develop a multi-objective mix-strategy DE (MOMSDE) algorithm. To motivate the diversity in mutation operations, MOMSDE incorporates the trick of a mixed-strategy that chooses a strategy from Eqs. (18)-(21) with equal probability (i.e., 0.25). It should be noted that MOMSDE adopts the traditional mutation strategies instead of guided counterparts because grouping individuals having two fitness values are unavailable. As the multi-objective problem incorporates Pareto dominance into the design of the optimization algorithm, we construct an optimal buffer in the proposed MOMSDE algorithm, which works in the following two aspects [46].

First, the buffer saves and updates the best non-dominated trail solutions in the current population, and outputs the Pareto optimal solutions when the algorithm converges. In the beginning, the buffer $B_{G=0}$ is initialized as the same as the current population. In the *selection*

operation, if the target solution $X_{i,G}$ does not dominate a trail solution $U_{i,G}$, the buffer is updated following three cases: (i) If any member of B_G dominates the trail solution $U_{i,G}$, $U_{i,G}$ is abandoned; (ii) If $U_{i,G}$ dominates a set of members from B_G, referred to as D_G, then $U_{i,G}$ is added to the current buffer B_G while the subset D_G is removed form B_G; (iii) If $U_{i,G}$ is non-dominated by any member of B_G, U_i is added to the optimal buffer. Detailed algorithm implementation of updating the optimal buffer is presented in Algorithm 3.

Algorithm 3 Algorithmic description of updating the optimal buffer.

if Any member $B_{i,G}$ of B_G dominates $U_{i,G}$ **then**
  $U_{i,G}$ is rejected.
end if
if $U_{i,G}$ dominates a set of members D_G from B_G **then**
  First D_G is removed from B_G as

$$B_G \leftarrow B_G / D_G.$$

  Then $U_{i,G}$ is added to B_G, as

$$B_G \leftarrow B_G \cup \{U_{i,G}\}.$$

end if
if $U_{i,G}$ is non-dominated by any member of B_G **then**
  $U_{i,G}$ is added to B_G, as

$$B_G \leftarrow B_G \cup \{U_{i,G}\}.$$

end if

Second, the buffer judges which solution survives to the next generation when the current target solution $X_{i,G}$ and $U_{i,G}$ are non-dominated with each other. To better explore and fulfill the Pareto front, we evaluate the crowding degree of $X_{i,G}$ and $U_{i,G}$ and the less crowded one survives. In this paper, the crowding degree is measured by the harmonic average distance [46], which is estimated based on the distances $d_1, \cdots, d_k$ of k-nearest neighbor distances around the target solution X_i. We choose $k = 2$ and the harmonic average distance d is defined as

$$d = \frac{2}{\frac{1}{d_1} + \frac{1}{d_2}}, \tag{27}$$

while higher value of harmonic average distance implies less crowding degree.

Detailed description of MOMSDE is presented in Algorithm 4. Same with the algorithm MGSDE, in MOMSDE, we generate different mutation and crossover factors for each individual according to the normal distribution, i.e., $\mathrm{F}_{i,G} \sim N(n_1, \sigma_1)$, $\mathrm{CR}_{i,G} \sim N(n_2, \sigma_2)$.

4. Numerical results

In this section, we test the proposed two-step robust control design approach on several one-qubit gates and the CNOT gate. All the numerical simulations are implemented on a MATLAB platform (version 9.10.0.1684407). The hardware environment is Intel(*R*)-Core(TM) i7-8700 CPU, dominant frequency @3.20GHz, and 16G(RAM).

Algorithm 4 Algorithmic description of MOMSDE.

Initialization: Set the generation index $G = 0$ and randomly initialize the individuals and the optimal buffer B_G.
repeat
  for $i = 1 : NP$ (each individual) **do**
    Mutation
    Select a mutation strategy based on Eqs. (19)-(22) to generate a mutation vector $V_{i,G}$.
    Crossover
    Generate a trail vector $U_{i,G}$ via crossover operation.
    Selection
    Compute the fitness $f_1(U_{i,G})$ and $f_2(U_{i,G})$.
    if $X_{i,G}$ dominates $U_{i,G}$ **then**
      $U_{i,G}$ is abandoned.
    end if
    if $U_{i,G}$ dominates $X_{i,G}$ **then**
      $X_{i,G} \leftarrow U_{i,G}$,
      $f_1(X_{i,G}) \leftarrow f_1(U_{i,G}), f_2(X_{i,G}) \leftarrow f_2(U_{i,G})$.
      Update the optimal buffer B_G with $U_{i,G}$ using **Algorithm 3**.
    end if
    if $X_{i,G}$ and $U_{i,G}$ are non-dominated with each other **then**
      Update the optimal buffer B_G with $U_{i,G}$ using **Algorithm 3**.
      Compare the crowding degree $d(X_{i,G})$ and $d(U_{i,G})$ of $X_{i,G}$ and $U_{i,G}$. The less crowded one survives to the next generation.
    end if
  end for
  $G \leftarrow G + 1$.
until $G = G_{\max,2}$
Output: The approximate Pareto set $B^* = B_{G_{\max,2}}$.

4.1. Robust control of one-qubit gates

We first study the control of three one-qubit gates including Hadamard gate (H), Phase gate (S), and $T_{\frac{\pi}{8}}$ gate, which are defined as follows:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, T_{\frac{\pi}{8}} = \begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix}. \tag{28}$$

The system Hamiltonian of a two-level system can be given as

$$H[t, \boldsymbol{u}] = \sigma_z + u_x(t)\sigma_x, \tag{29}$$

where

$$\sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \tag{30}$$

are Pauli matrices and $u_x(t)$ represents the external control fields. Suppose that parameter disturbances exist in both of the free Hamiltonian and control Hamiltonian, and the system

Table 2
Training fitness of the fidelity optimization step for the Hadamard gate (H), Phase gate (S), and $T_{\frac{\pi}{8}}$ gate.

Algorithm	Parameters	Fitness (average fidelity)		
		H	S	T
DE	F = 0.9, CR = 0.1	0.99443	0.99425	0.99702
GA	11-bit encoding, $P_c = 0.8$, $P_m = 0.005$	0.99227	0.99178	0.99287
MSDE	F = $N(0.5, 0.3)$, CR = $N(0.5, 0.1)$	0.99632	0.99470	0.99623
MGSDE	F = $N(0.5, 0.3)$, CR = $N(0.5, 0.1)$	0.99968	0.99988	0.99992
GRAPE	the learning rate $\alpha = 0.05$	0.99959	0.99959	0.99961

Hamiltonian is described as

$$H[t; \boldsymbol{u}, \boldsymbol{\epsilon}] = (1 + \omega)\sigma_z + (1 + \theta)u_x(t)\sigma_x. \tag{31}$$

We evenly choose five samples for each uncertainty parameter. The fitness function for the first step of optimization is set as the average fidelity based on the 25 samples, that is

$$J_1(\boldsymbol{u}) = \frac{1}{25}\sum_{k=1}^{25} F(U_F, \boldsymbol{\epsilon}_k). \tag{32}$$

The control $u_x(t)$ are divided into $D = 40$ piece-wise constant segments with the boundary of $u_j \in [-5, +5]$, $j = 1, \cdots, D$. We set the disturbance boundary $E = 0.2$ and the transfer time $T = 8$. To demonstrate the efficiency of the proposed MGSDE, we implement another four algorithms, including standard differential evolution (DE), mixed-strategy differential evolution (MSDE), genetic algorithm (GA) and the gradient method. The population size is 60 and the learning iterations are set as 20,000 for DE, MSDE, MSGDE and GA. The gradient method utilizes 500,000 learning iterations with the learning rate $\alpha = 0.05$.

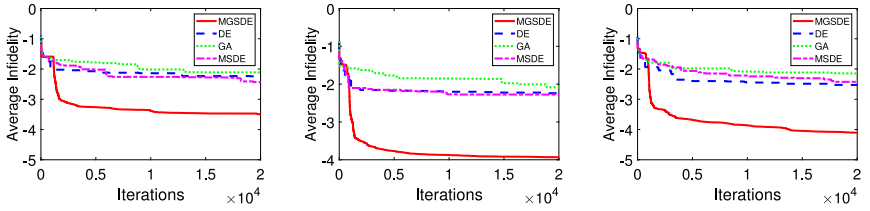
Detailed parameters and results are presented in Table 2. The learning curves are shown as in Fig. 4, where the average infidelity is referred to as $\log_{10}(1 - J_1(\boldsymbol{u}))$. Among all the evolutionary algorithms, MGSDE converges fastest and achieves the highest performance with average fidelities of around 99.99%. The gradient method converges and achieves average fidelity around 99.96%, but it performs no better than MGSDE.

The robustness optimization step is a multi-objective task aiming at

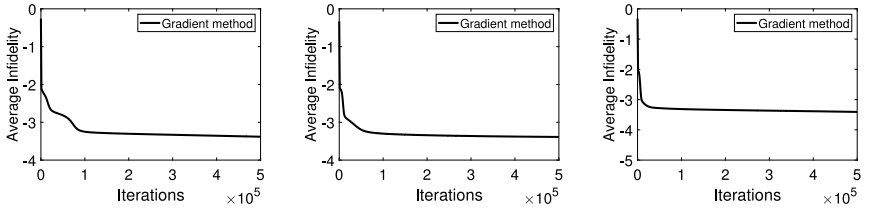
$$\begin{aligned} &\max_{\boldsymbol{u}} f_1(\boldsymbol{u}) = \frac{1}{25}\sum_{k=1}^{25} F(\boldsymbol{u}, \boldsymbol{\epsilon}_k); \\ &\min_{\boldsymbol{u}} f_2(\boldsymbol{u}) = \frac{1}{24}\sum_{k=1}^{25} |L(\boldsymbol{u}, \boldsymbol{\epsilon}_k) - \mu(\boldsymbol{u}, \boldsymbol{\epsilon}_k)|^2. \end{aligned} \tag{33}$$

We implement the MOMSDE algorithm with the population size $NP = 60$ and 50,000 learning iterations. The obtained Pareto fronts are shown as in Fig. 5. It should be noted that we have only stored solutions achieving average fidelity no less than 0.999 to satisfy the quantum error correction threshold (e.g., 0.001 for the gate error) [47].

The Pareto fronts show an intuitive trend that controls with higher average fidelity possess higher infidelity variance, which denotes the conflict between the optimal performance and robustness. We choose control strategies with the median performance from the obtained Pareto sets to make a trade-off. The detailed information of the chosen control strategies is listed in Table 3. We implement the standard SLC method [12] for comparison with only

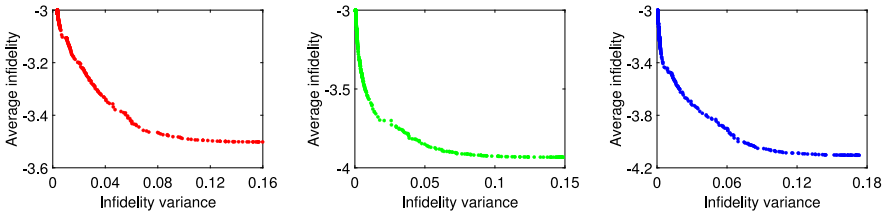


(a) MGSDE, MSDE, DE and GA for Hadamard gate. (b) MGSDE, MSDE, DE and GA for Phase gate. (c) MGSDE, MSDE, DE and GA for $T_{\frac{\pi}{8}}$ gate.



(d) The gradient method for Hadamard gate. (e) The gradient method for Phase gate. (f) The gradient method for $T_{\frac{\pi}{8}}$ gate.

Fig. 4. Training curves of the fidelity optimization step.



(a) Pareto solution set of Hadamard gate. (b) Pareto solution set of Phase gate. (c) Pareto solution set of $T_{\frac{\pi}{8}}$ gate.

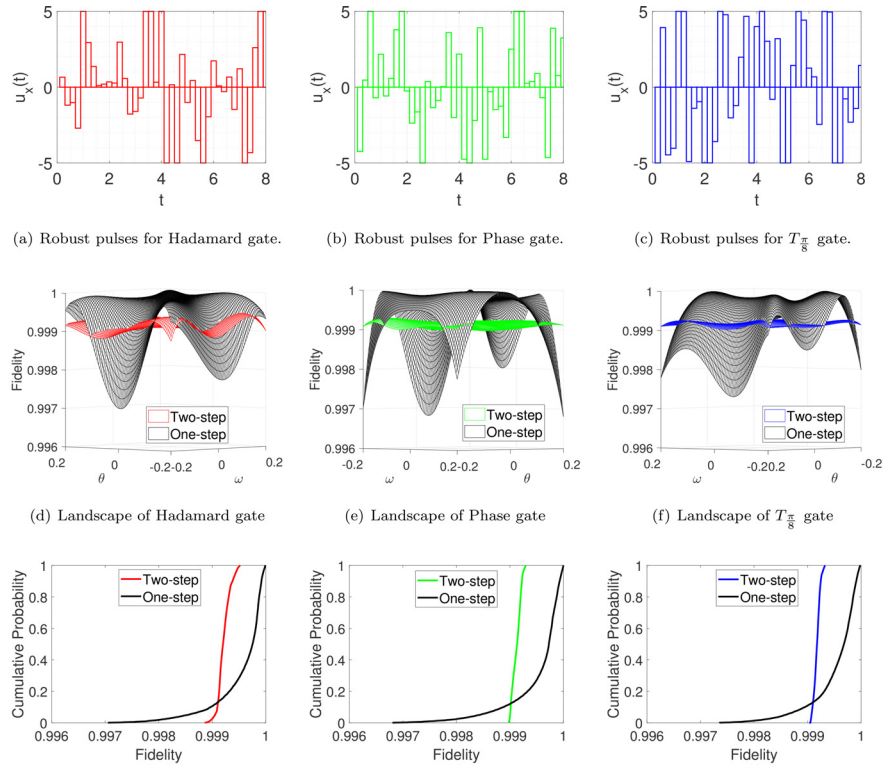
Fig. 5. Pareto front.

Table 3
The training fitness of the applied control pulses.

Fitness	Case		
	H	S	T
Infidelity variance	0.0058	0.0007	0.0009
Average fidelity	0.9992	0.9991	0.9992

one step that optimizes the average fidelity. The average fidelity goals for the standard SLC method are the same as the two-step approach.

We test these robust control strategies with uniformly distributed uncertainty samples and present the control landscapes in Fig. 6 (d)-(f). Here, the control landscape denotes the mapping from uncertainty parameters ω and θ to the performance measure (e.g., quantum gate fidelity). It can be observed that the gate fidelity under the one-step control varies dramatically and is significantly suppressed to be below 0.999 (at around 0.997) at fluctuation boundary of ω and θ. In contrast, the landscapes associated with the two-step approach are flat. We also calculate the gate fidelity distributions and present them in Fig. 6 (g)-(i). Notably, the two-



(a) Robust pulses for Hadamard gate. (b) Robust pulses for Phase gate. (c) Robust pulses for $T_{\frac{\pi}{8}}$ gate.

(d) Landscape of Hadamard gate (e) Landscape of Phase gate (f) Landscape of $T_{\frac{\pi}{8}}$ gate

(g) Cumulative probability of gate fidelity of Hadamard gate (h) Cumulative probability of gate fidelity of Phase gate (i) Cumulative probability of gate fidelity of $T_{\frac{\pi}{8}}$ gate

Fig. 6. Robust controls and testing results.

step approach remarkably improves robustness as it makes the gate fidelity less insensitive to fluctuations and enhances the worst-case fidelity (nearly no less than 0.999).

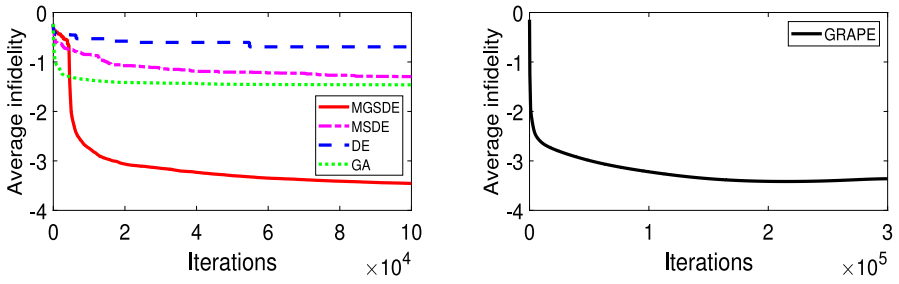
4.2. Robust control of CNOT gate

CNOT gate acts on two qubits that flips the second qubit if and only if the first qubit is at state $|1\rangle$, whose matrix representation is given as

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{34}$$

The system Hamiltonian is

$$\begin{aligned} H[t] &= H_0 + \sum_{i=1}^{m} u_m(t) H_m \\ &= \sigma_1^z \otimes \sigma_2^z + u_1(t)\sigma_1^x \otimes \mathbb{I}_2 + u_2(t)\mathbb{I}_2 \otimes \sigma_2^x \\ &\quad + u_3(t)\sigma_1^y \otimes \mathbb{I}_2 + u_4(t)\mathbb{I}_2 \otimes \sigma_2^y, \end{aligned} \tag{35}$$



(a) Learning curves via MGSDE, MSDE, DE, and GA. (b) Learning curve via GRAPE.

Fig. 7. Learning curves in the fidelity optimization step for the CNOT gate.

Table 4
Training performance of CNOT gate via GA, DE, MSDE, MGSDE, and Gradient method.

Algorithm	Parameter	Fitness(average fidelity)
DE	F = 0.9, CR = 0.1	0.79773
MSDE	F = $N(0.5, 0.3)$, CR = $N(0.5, 0.1)$	0.94945
MGSDE	F = $N(0.5, 0.3)$, CR = $N(0.5, 0.1)$	0.99965
GA	11-bit encoding, $P_c = 0.8$,$P_m = 0.005$	0.96553
GRAPE	learning rate $\alpha = 0.02$	0.99956

where $\mathbb{I}_2$ is the 2×2 identity matrix and $\sigma_k^{x,y,z}(k = 1, 2)$ represents the Pauli operator associated with the kth qubit. Considering parameter uncertainties, the system Hamiltonian is given as

$$H[t; \boldsymbol{u}, \boldsymbol{\epsilon}] = (1 + \omega)H_0 + (1 + \theta)\sum_{m=1}^{4} u_m(t)H_m, \tag{36}$$

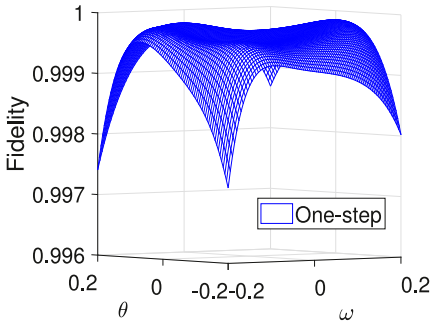
where ω and θ are assumed to be uniformly distributed over $[-E, E]$.

We set the disturbance bound as $E = 0.2$ and set $T = 8$. The control fields $\{u_m(t), m = 1, 2, 3, 4\}$ are evenly divided into $D = 40$ piece-wise constant segments and bounded as $u_m(t) \in [-5, +5]$.

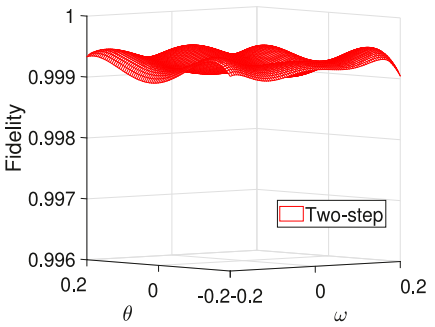
The fidelity optimization step aims at improving the average performance. We evenly select three samples for each uncertainty parameter in the first step, thus $n = 9$. The training fitness function in the fidelity optimization step is given as

$$J_1(\boldsymbol{u}) = \frac{1}{9}\sum_{k=1}^{9} F(\boldsymbol{u}, \boldsymbol{\epsilon}_k). \tag{37}$$

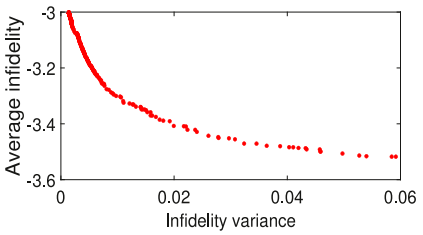
We implement the five algorithms of DE, GA, MSDE, MGSDE and the gradient method. The population size is set as 60. The training curves are presented in Fig. 7 and detailed results are listed in Table 4. The MGSDE algorithm achieves the highest training performance among all the five algorithms, as it achieves the average fidelity $\bar{F} = 0.99965$. It is clear that MGSDE is superior to the other three evolutionary algorithms and is comparable with the gradient method. After 100,000 iterations, the learning curve of MGSDE still shows potential to achieve higher performance, and we give another 100,000 iterations to MGSDE and finally achieve a training performance of 0.99974.



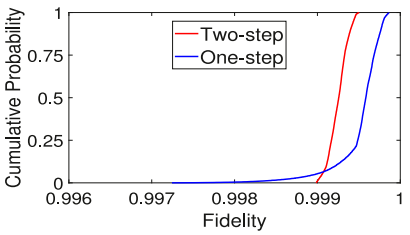
(a) Control landscape of one-step method.



(b) Control landscape of two-step robust method.



(c) Pareto solution set obtained via MOMSDE.



(d) Cumulative probability distribution of gate fidelity.

Fig. 8. Testing results of CNOT gate.

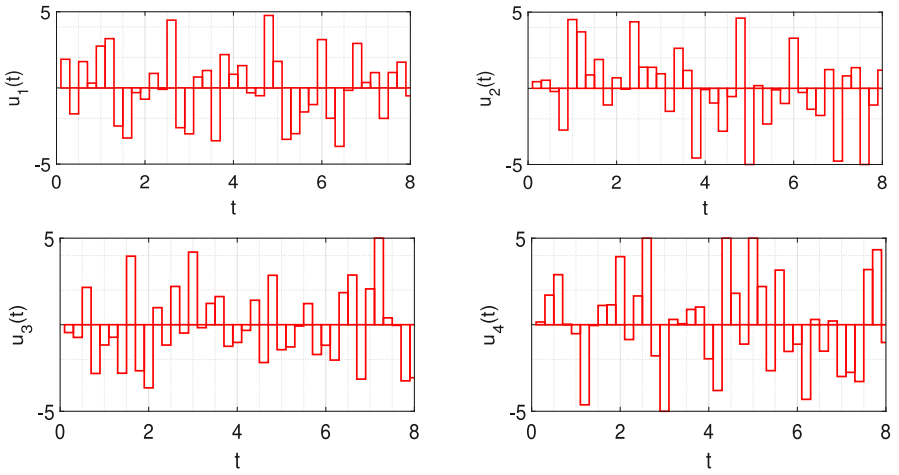


Fig. 9. Robust control pulses obtained via the two-step robust approach for CNOT gate.

For the robustness optimization step, the objective includes the average fidelity and the infidelity variance based on 25 samples that are evenly sampled. The MOMSDE algorithm is implemented with 50,000 iterations, which provides a front set consisting of 833 non-dominated solutions that are presented in Fig. 8 (c). To make a balance between performance and robustness, we choose a control strategy with the average fidelity $\bar{F} = 0.9992$ and the infidelity variance $\sigma^2[L] = 0.0031$. To make comparison, we implement the one-step approach, which achieves the average fidelity $\bar{F} = 0.9992$. The testing results and robust control pulses

are presented in Fig. 8 and Fig. 9, respectively. For the landscape of the one-step approach, although the gate fidelity is high in most cases, it drops below 0.999 at the fluctuation bound.

As shown in Fig. 8 (b), our two-step method makes the gate fidelity less insensitive to bounded uncertainties and maintains high fidelity under the worst disturbance. The numerical results demonstrate that optimizing infidelity variance can remarkably reduce the fitness dispersion for improved robustness.

5. Conclusion

In this paper, we propose a two-step robust control approach using improved differential evolution algorithms for quantum gates that first optimizes the fidelity performance and then enhances control robustness. We propose an improved MGSDE algorithm for the first fidelity optimization step, and an MOMSDE algorithm for the second robustness optimization step with an optimal buffer and mixed-strategies to achieve a set of non-dominated optimal solutions. Numerical results show that our two-step robust approach could improve the control robustness against parameter disturbances while maintaining high precision. We will focus on developing more efficient learning algorithms (e.g., deep reinforcement learning algorithms [48]) to enhance control robustness in our future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Algorithm description of Genetic Algorithm

The genetic algorithm (GA) is a kind of population-based search algorithm [49]. In particular, GA builds a population S using NP individuals, while each individual solution is encoded as a chromosome. As shown in Algorithm 5, for each iteration, GA computes the

Algorithm 5 Genetic Algorithm.

Set generation number $G = 0$
Initialize a population S_G consisting of NP randomly generated chromosomes $X_{i,G}$
for G = 1 to $G_{\max}$ **do**
  for i=1 to NP **do**
    Decode the chromosome and compute the fitness $J_{i,G} = J(X_{i,G})$;
  end for
  Following the descend direction of $J_{i,G}$, constitute the sub-population $S_{1,G}$ and $S_{2,G}$;
  Perform crossover and mutation operations on $S_{2,G}$;
  $G \leftarrow G + 1$
  Update the population as $S^G \leftarrow S_{1,G} \cup S_{2,G}$;
end for
Output the best solution X^*

fitness values, and ranks the individuals and builds two sub-populations S_1 and S_2. With a crossover rate P_c, top $[NP(1 - P_c)]$ chromosomes are selected to constitute the sub-set S_1,

which directly survives to the next generation. Then, GA chooses $NP * P_c$ chromosomes with probability $P_i = \frac{J_i}{\sum_{i=1}^{NP} J_i}$ to constitute S_2. To renew chromosomes in S_2, GA will perform crossover operations with P_c and implement mutation operations with mutation factor P_m. Finally, S_1 and S_2 jointly constitute the new generation.

Appendix B. Algorithm description of Gradient-based method

Algorithm 6 gives description of the gradient-based method (GRAPE). In particular, the

Algorithm 6 Gradient-based Algorithm.

Iteration index $k = 1$;
Randomly initialize the control parameters $\boldsymbol{u}_0 = [u_1, \cdots, u_N]$,where $u_j = u_{\min} + \text{rand}(0, 1) \cdot (u_{\max} - u_{\min}), j = 1, \cdots, N$
for $k = 1$ to $K_{\max}$ **do**
  for $j = 1$ to D **do**
    Compute and record the jth propagator U_j
  end for
  Compute the gradient information $\frac{\delta J(\boldsymbol{u}^k)}{\delta \boldsymbol{u}^k}$
  Update control parameter $\boldsymbol{u}^{k+1} \leftarrow \boldsymbol{u}^k + \alpha \frac{\delta J(\boldsymbol{u}^k)}{\delta \boldsymbol{u}^k}$
end for
Optimal control parameters $\boldsymbol{u}^* = \boldsymbol{u}^{K_{\max}}$

piece-wise constant control field $\boldsymbol{u}$ has D segments. U_j is the jth propagator as $U(T) = U_N \cdots U_j \cdots U_1$. The gradient information is given as [21]

$$\frac{\delta J(u_j)}{\delta u_j} = -2Re\{\langle B_j | i\Delta t H A_j \rangle \langle A_j | B_j \rangle\}$$

and operators A_j, B_j are defined as $A_j = U_j \cdots U_1$ and $B_j = A_j U(T)^\dagger U_f$.

CRediT authorship contribution statement

Shouliang Hu: Methodology, Software, Writing – original draft. **Hailan Ma:** Data curation, Visualization. **Daoyi Dong:** Investigation, Writing – review & editing. **Chunlin Chen:** Conceptualization, Writing – review & editing, Supervision.

References

[1] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. 41 (2) (1999) 303–332.
[2] F. Schmidt-Kaler, H. Häffner, M. Riebe, S. Gulde, G.P. Lancaster, T. Deuschle, C. Becher, C.F. Roos, J. Eschner, R. Blatt, Realization of the cirac–zoller controlled-not quantum gate, Nature 422 (6930) (2003) 408–411.
[3] E. Knill, Quantum computing, Nature 463 (7280) (2010) 441–443.
[4] D. Dong, I.R. Petersen, Quantum estimation, control and learning: opportunities and challenges, Annual Reviews in Control, in press (2022), doi:10.1016/j.arcontrol.2022.04.011.
[5] D. Hocker, C. Brif, M.D. Grace, A. Donovan, T.-S. Ho, K.M. Tibbetts, R. Wu, H. Rabitz, Characterization of control noise effects in optimal quantum unitary dynamics, Phys. Rev. A 90 (6) (2014) 062309.
[6] M. Dahleh, A. Peirce, H. Rabitz, Optimal control of uncertain quantum systems, Phys. Rev. A 42 (3) (1990) 1065.

[7] H.-J. Ding, R.-B. Wu, Robust quantum control against clock noises in multiqubit systems, Phys. Rev. A 100 (2) (2019) 022302.
[8] D. Dong, C. Chen, R. Long, B. Qi, I.R. Petersen, Sampling-based learning control for quantum systems with hamiltonian uncertainties, in: 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 1924–1929.
[9] R.L. Kosut, M.D. Grace, C. Brif, Robust control of quantum gates via sequential convex programming, Phys. Rev. A 88 (5) (2013) 052326.
[10] C. Chen, R. Long, B. Qi, D. Dong, Sampling-based learning control of quantum systems with bounded inputs and uncertainties via path planning, in: 2013 Australian Control Conference, IEEE, 2013, pp. 121–126.
[11] C. Chen, D. Dong, B. Qi, I.R. Petersen, H. Rabitz, Sampling-based learning control for quantum discrimination and ensemble classification, in: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 2014a, pp. 880–885.
[12] C. Chen, D. Dong, R. Long, I.R. Petersen, H.A. Rabitz, Sampling-based learning control of inhomogeneous quantum ensembles, Phys. Rev. A 89 (2) (2014b) 023402.
[13] C. Chen, R. Long, B. Qi, D. Dong, Sampling-based learning control of quantum systems via path planning, IET Control Theory Appl. 8 (15) (2014c) 1513–1522.
[14] M.A. Mabrok, D. Dong, C. Chen, I.R. Petersen, Robust entanglement control between two atoms in a cavity using sampling-based learning control, in: 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 5802–5807.
[15] C. Wu, C. Chen, B. Qi, D. Dong, Robust quantum operation for two-level systems using sampling-based learning control, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2015, pp. 2043–2048.
[16] H. Ma, C. Chen, D. Dong, Differential evolution with equally-mixed strategies for robust control of open quantum systems, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2015, pp. 2055–2060.
[17] Y. Sun, H. Ma, C. Wu, C. Chen, D. Dong, Ensemble control of open quantum systems using differential evolution, in: 2015 10th Asian Control Conference (ASCC), IEEE, 2015, pp. 1–6.
[18] D. Dong, C. Chen, B. Qi, I.R. Petersen, F. Nori, Robust manipulation of superconducting qubits in the presence of fluctuations, Sci. Rep. 5 (1) (2015a) 1–6.
[19] D. Dong, M.A. Mabrok, I.R. Petersen, B. Qi, C. Chen, H. Rabitz, Sampling-based learning control for quantum systems with uncertainties, IEEE Trans. Control Syst. Technol. 23 (6) (2015b) 2155–2166.
[20] D. Dong, C. Wu, C. Chen, B. Qi, I.R. Petersen, F. Nori, Learning robust pulses for generating universal quantum gates, Sci. Rep. 6 (1) (2016) 1–9.
[21] C. Wu, B. Qi, C. Chen, D. Dong, Robust learning control design for quantum unitary transformations, IEEE Trans. Cybern. 47 (12) (2016) 4405–4417.
[22] D. Dong, X. Xing, H. Ma, C. Chen, Z. Liu, H. Rabitz, Learning-based quantum robust control: algorithm, applications, and experiments, IEEE Trans. Cybern. 50 (8) (2020) 3581–3593.
[23] R. Chakrabarti, H. Rabitz, Quantum control landscapes, Int. Rev. Phys. Chem. 26 (4) (2007) 671–735.
[24] C. Joe-Wong, T.-S. Ho, R. Long, H. Rabitz, R. Wu, Topology of classical molecular optimal control landscapes in phase space, J. Chem. Phys. 138 (12) (2013) 124114.
[25] L. Van Damme, Q. Ansel, S. Glaser, D. Sugny, Robust optimal control of two-level quantum systems, Phys. Rev. A 95 (6) (2017) 063403.
[26] X. Ge, R.-B. Wu, Risk-sensitive optimization for robust quantum controls, Phys. Rev. A 104 (1) (2021) 012422.
[27] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, S.J. Glaser, Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms, J. Magn. Reson. 172 (2) (2005) 296–305.
[28] R.-B. Wu, H. Ding, D. Dong, X. Wang, Learning robust and high-precision quantum controls, Phys. Rev. A 99 (4) (2019) 042327.
[29] X. Ge, H. Ding, H. Rabitz, R.-B. Wu, Robust quantum control in games: an adversarial learning approach, Phys. Rev. A 101 (5) (2020) 052317.
[30] D. Dong, Learning control of quantum systems, In Encyclopedia of Systems and Control, J. Baillieul, T. Samad (eds.), Springer-Verlag London Ltd, (2021) 1090–1096, https://doi.org/10.1007/978-1-4471-5102-9100161-1 (????).
[31] R.S. Judson, H. Rabitz, Teaching lasers to control molecules, Phys. Rev. Lett. 68 (10) (1992) 1500.
[32] V. Manu, A. Kumar, Singlet-state creation and universal quantum computation in nmr using a genetic algorithm, Phys. Rev. A 86 (2) (2012) 022324.
[33] V.C. Gregoric, X. Kang, Z.C. Liu, Z.A. Rowley, T.J. Carroll, M.W. Noel, Quantum control via a genetic algorithm of the field ionization pathway of a rydberg electron, Phys. Rev. A 96 (2) (2017) 023403.

[34] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[35] B. Hegerty, C.-C. Hung, K. Kasprak, A comparative study on differential evolution and genetic algorithms for some combinatorial problems, in: Proceedings of 8th Mexican International Conference on Artificial Intelligence, volume 9, 2009, p. 13.
[36] E. Zahedinejad, J. Ghosh, B.C. Sanders, High-fidelity single-shot toffoli gate via quantum control, Phys. Rev. Lett. 114 (20) (2015) 200502.
[37] E. Zahedinejad, J. Ghosh, B. Sanders, Designing high-fidelity single-shot three-qubit gates: a machine-learning approach, Phys. Rev. Appl. 6 (5) (2016) 054005.
[38] X. Yang, J. Li, X. Peng, An improved differential evolution algorithm for learning high-fidelity quantum controls, Sci. Bull. 64 (19) (2019) 1402–1408.
[39] X.-D. Yang, C. Arenz, I. Pelczer, Q.-M. Chen, R.-B. Wu, X. Peng, H. Rabitz, Assessing three closed-loop learning algorithms by searching for high-quality quantum control pulses, Phys. Rev. A 102 (6) (2020) 062605.
[40] A.W. Mohamed, A.K. Mohamed, Adaptive guided differential evolution algorithm with novel mutation for numerical optimization, Int. J. Mach. Learn. Cybern. 10 (2) (2017) 253–277.
[41] B. Rowland, J.A. Jones, Implementing quantum logic gates with gradient ascent pulse engineering: principles and practicalities, Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci. 370 (1976) (2012) 4636–4650.
[42] P. Ngatchou, A. Zarei, A. El-Sharkawi, Pareto multi objective optimization, in: Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, IEEE, 2005, pp. 84–91.
[43] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2008) 398–417.
[44] R. Mallipeddi, P.N. Suganthan, Q.-K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft. Comput. 11 (2) (2011) 1679–1696.
[45] M.G. Omran, A. Salman, A.P. Engelbrecht, Self-adaptive differential evolution, in: International Conference on Computational and Information Science, Springer, 2005, pp. 192–199.
[46] V.L. Huang, P.N. Suganthan, A.K. Qin, S. Baskar, Multiobjective differential evolution with external archive and harmonic distance-based diversity measure, SchoolElectric. Electron. Eng. Nanyang Technol. Univ. Tech. Rep. (2005).
[47] C.H. Bennett, D.P. DiVincenzo, J.A. Smolin, W.K. Wootters, Mixed-state entanglement and quantum error correction, Phys. Rev. A 54 (5) (1996) 3824.
[48] Z. Ren, D. Dong, H. Li, C. Chen, Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2216–2226.
[49] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimed. Tools Appl. 80 (5) (2021) 8091–8126.