2022 IEEE International Conference on
Systems, Man, and Cybernetics (SMC)

A Modified Deep Q-Learning Algorithm for Optimal and Robust Quantum Gate Design of a Single Qubit System*

Omar Shindi[1], Qi Yu[1,2], Parth Girdhar[1], Daoyi Dong[1]

2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-5258-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SMC53654.2022.9945165

***Abstract*—Precise and resilient quantum gate design is important for the building of quantum devices. In this paper, we consider the optimal and robust quantum gate design problem for three classes of two-level quantum systems. The aim is to construct quantum gates in a given fixed time with limited control resources. A modified dueling deep Q-learning (MDuDQL) is employed for the optimal and robust gate design problem. To improve the performance of the classical DuDQL method, we propose a unique semi-Markov DuDQL algorithm based on a modified action selection procedure, modified replay memory, and soft update procedure. The proposed algorithm outperforms ordinary DuDQL in terms of discovering global optimal or near-global optimal control protocols and faster convergence to a better policy. Moreover, the modified DuDQL agent shows improved performance in finding robust control protocols which achieve high-fidelity quantum gate design for varying uncertainties in a certain range. The effectiveness of the proposed algorithm for the optimal and robust gate design problems has been illustrated by numerical results.**

I. Introduction

Quantum gate design is essential in many areas such as quantum computation [1]- [3] and communication [4], [5]. The main task of quantum gate design is to find out a desired unitary operation [6]. Achieving a good quality quantum gate design can be challenging, especially when uncertainties of the system dynamics and limits of control resources are introduced to the problem [7], [8].

There are various methods for quantum gate steering [9], [10]. Optimal quantum gate design formulates the task as an optimization problem which aims to minimize a specified cost function by tuning the control parameters, given restrictions on control resources [11], [12]. Several algorithms have been proposed for solving optimal quantum control problems [13]- [20], such as optimal control gradient-based methods for quantum gate design and state preparation [19], [20]. In a realistic situation, the quantum system is unavoidably vulnerable to noise and incomplete knowledge [21]. As a result, the optimal solution may not be feasible without extensive calibration in a real-world experiment. In order to increase the feasibility of the optimal control solution, the noise and disturbances can be characterized as uncertain parameters in the model of quantum systems. Then, robust control methods can be used to find the single shot control protocol that achieves the control goal on a restricted range of uncertainty [22]. Various approaches [23]- [31] such as enhanced gradient based algorithms [26], [27], differential evolution algorithms [28] and sampling based learning method [29], [30] have been proposed for robust gate design. However, gradient-based algorithms are sensitive to initial guesses and local optima, which limit the efficiency of these algorithms. Furthermore, with limited control resources, such as time discretization and fixed amplitude of the control fields, it is difficult to obtain such high-quality gates.

Deep reinforcement learning (DRL) [32], [33], has recently had great success and has demonstrated human-level capabilities in a variety of control activities, such as mastering video games [34], Go game [35], [36], and controlling nuclear fusion plasma for a nuclear power plant [37]. With such excellent performance, DRL has attracted attention in quantum technology research and has been adopted for challenging problems such as quantum control [38]- [42], quantum error correction [43], quantum parameters estimation [44] and quantum Hamiltonian engineering [45]. However, designing high-fidelity and robust quantum gates is still difficult, especially when control resources are limited. In general, DRL approaches suffer from the exploration-exploitation trade-off problem [46]. A balance needs to be achieved between exploration, which discovers unknown potentially highly rewarding regions, and exploitation of profitable regions that have already been discovered in the environment. A proper exploration-exploitation balance can help the DRL agent avoid being stuck into local optimum solutions.

In this work, we propose a semi-Markov Modified Dueling DQL (MDuDQL) algorithm for addressing the optimal and robust gate design problem based on a modified action selection mechanism, improved replay memory and soft update procedure. The modified action selection strategy would accelerate the discovery of the best results. And the soft update procedure would help the MDuDQL agent to achieve a better convergence behaviour by avoiding catastrophic updates for the policy. Furthermore, the delayed reward function and improved replay memory would help in obtaining a better control policy. The MDuDQL approach is compared with traditional dueling DQL (DuDQL) based Markov and semi-Markov methods. Numerical results on three basic quantum gates reveal that MDuDQL outperforms the regular DuDQL in terms of achieving global optimal results with better fidelity and robustness.

The remainder of this paper is organized as follows. Section II describes the optimal and robust quantum gate design problems. Section III presents background for deep

*This work was supported by the Australian Research Council's Discovery Projects funding scheme under Project DP190101566 and the U. S. Office of Naval Research Global under Grant N62909-19-1-2129.

[1] School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (email: omar.shindi.ca@gmail.com, drparthgirdhar@gmail.com, daoyidong@gmail.com).

[2] Center for Quantum Dynamics, Griffith University, Brisbane, Queensland 4111, Australia (email: yuqivicky92@gmail.com).

RL and an introduction to the improved version of DuDQL. The proposed improvements, the modified action selection procedure, the semi-Markov procedure, and the soft update process are analyzed and explained in Section IV. The numerical results and discussion are provided in Section V. Concluding remarks are given in Section VI.

II. Quantum gate design problem

The size of a unitary matrix of single qubit gate is 2x2. Therefore, a n-qubit gate can be represented by unitary matrix U of size 2^nx2^n in complex Hilbert space $\mathscr{H}$. The quantum gate design problem aims to develop a control protocol $\mathbf{A} = \{u_1, u_2, ...\}$ which drives a gate U_0 at time $t = 0$ to a target unitary U_T at time $t = T$. At time step t, the approximate unitary U_t can be calculated using the following time evolution equation.

$$U_t = e^{-iH(u_t)dt}U_{t-1}, \tag{1}$$

where i is unit imaginary number, $H(u_t)$ is the Hamiltonian of the quantum system, U_{t-1} is the unitary operator at the previous time step $t-1$ and dt is the time period of applied external control pulse u_t. The Hamiltonian $H(u_t)$ can be written as

$$H(u_t) = H_0 + H_c(u_t), \tag{2}$$

where H_0 and $H_c(u_t)$ are the free Hamiltonian and the control Hamiltonian, respectively. $H_c(u_t)$ represents the part of H that can be engineered by the external control field u_t.

In this work, we consider the Hamiltonian of a two-level quantum system as

$$H_0 = \sigma_z, \tag{3}$$

$$H_c(u_t) = u_t\sigma_x, \tag{4}$$

where

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{5}$$

are the Pauli operators of the single-qubit system. The final unitary operator for the applied N-steps control protocol $\mathbf{A} = \{u_1, u_2, ..., u_N\}$ and the total evolution time T can be found as

$$U_f = e^{-iH(u_N)dt}e^{-iH(u_{N-1})dt}...e^{-iH(u_1)dt}U_0, \tag{6}$$

where the pulse duration is assumed to be fixed equal to $dt = T/N$ and the initial unitary operator U_0 is considered equal to the identity operator $\mathbb{I}_{2x2}$.

$$\mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7}$$

A. Optimal gate design problem

The optimal quantum control aims to determine the proper control protocol $\mathbf{A} = \{u_1, u_2, ..., u_N\}$ that minimizes the objective function J, while constraints can be placed on the total evolution time T and the amplitude of the available control pulses. The quantum gate design problem aims to steer the unitary operator U_f to the target operator U_T at the end of the evolution time T. The objective function is often chosen to be the infidelity [47] that can be measured as

$$J = 1 - \left| \frac{\mathrm{Tr}\{U_f^\dagger U_T\}}{2^n} \right|^2. \tag{8}$$

In this work, we consider the design of three classes of fundamental gates: Hadamard-gate (H), T-gate (T) and S-gate (S), for the quantum system described in equations (3) and (4).

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \mathrm{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}, S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{9}$$

This work aims to find a proper control protocol sequence of $u \in \{-4, 4\}$ on a discrete action space to achieve the control task with high fidelity. The total evolution times for the gates H, S, and T are $T \in \{1.5, 1.1, 1.3\}$, respectively.

B. Robust gate design problem

Theoretically, optimal control solutions have proven effective for high-fidelity quantum gate design, but in reality they require extensive calibration to be performed. This problem occurs because of difficulty to construct a highly characterized dynamic model for the quantum system because of the sensitivity to the surrounding environment. Robust gate design solves this issue and aims to find a control protocol that achieves the design goals for different possible values of uncertain parameters. To test the effectiveness of the proposed improvements for robust gate design, a Hamiltonian with the following model imperfection has been considered

$$H(\vec{\epsilon}) = (1 + \epsilon_1)\sigma_z + (1 + \epsilon_2)u\sigma_x, \tag{10}$$

where $\vec{\epsilon}$ is a pair of uncertain parameters and the uncertain range is set to be $|\epsilon_j| \leq 0.1$ for $j = 1, 2$. Further, it is general to assume that both parameters are uniformly distributed in the given range as:

$$\epsilon_1, \epsilon_2 \in \{-0.1, -0.1 + \frac{0.2m}{p}, 0.1 | m = 2, 3, ..., p-1\}. \tag{11}$$

Thus, the space that contains all possible pairs of uncertain parameters is of size $(p+1)^2$. The goal is to find a robust control protocol within the total evolution time T and the number of steps N that results in the final unitary operators $U_f(H(\vec{\epsilon}))$ for the majority of elements of $\vec{\epsilon}$ as close as possible to the target unitary U_T. The final unitary operators $U_f(H(\vec{\epsilon}))$ can be computed as

$$U_f(H(\vec{\epsilon})) = e^{-iH(u_N,\vec{\epsilon})dt}e^{-iH(u_{N-1},\vec{\epsilon})dt}...e^{-iH(u_1,\vec{\epsilon})dt}U_0 \tag{12}$$

The infidelities for the final unitary operators $U_f(H(\vec{\epsilon})) \in \{U_{f_1}, U_{f_2}..., U_{f_{(p+1)^2}}\}$ can be computed as in (8). Unlike the optimal gate design, we will have a vector of infidelities $\vec{J} = (J_1, J_2, ..., J_{(p+1)^2})$ that describes the quality of the applied control protocol in each element of $\vec{\epsilon}$. The robust gate design problem can be solved as an optimization problem with the aim of minimizing an objective function $\bar{J}$ by building the appropriate control protocol. In this work, the objective function $\bar{J}$ is chosen as follows

$$\bar{J} = w\max(\vec{J}) + (1 - w)\mathrm{Avg}(\vec{J}), \tag{13}$$

$$\mathrm{Avg}(\vec{J}) = \frac{\sum_{j=1}^{(p+1)^2} J_j}{(p+1)^2}. \tag{14}$$

$$\max(\vec{J}) = \max(\vec{J} \in \{J_1, J_2, ..., J_{(p+1)^2}\}) \tag{15}$$

The $\vec{J}$ is the vector of infidelities corresponding to different uncertainty values, and w is a weight parameter to balance between the average infidelity achieved and the worst infidelity. The total evolution time of the robust gate design problem is the same for the optimal gate design problem in section II-A except for the T-gate, which is $T = 3.9$.

III. Deep Q-learning (DQL) Algorithm

DQL is a temporal difference Reinforcement Learning (RL) approach that uses Neural Networks (NNs) to approximate the Q-value function in a finite Markov decision process (MDP). Thus, the DQL agent is able to learn the control behavior without prior knowledge. It is widely used for complex, high-dimensional problems, such as video games [48], [49].

The environment and the RL agent are the two main components of the RL process. The environment is usually expressed as MDP that can be described by a tuple $\{O, \mathcal{A}, r\}$, where O is the environment observation, $\mathcal{A}$ is the action space and r is the reward. The RL agent is a policy or a value function that maps between the environment observation O and the action space $\mathcal{A}$ in the case of discrete action space. Concretely, the DQL agent contains two Q-value functions represented by two similar architecture neural networks: the target-network with weights θ_T, and the value-network with weights θ_V. The Q-value function $Q(O_t, a_t, \theta)$ is an evaluation of the expected return for the chosen action a_t in a state O_t based on the function weights θ at the time step t.

The DQL agent interacts with the environment $\mathcal{E}$ in a series of episodes to learn to make decisions. Each episode is a sequential decision-making process for N discrete time steps. At each time step $t \in [0, 1, ..., N-1]$, the DQL agent receives an observation of the environment O_t and then chooses an action a_t to interact with the environment. Methods such as $\varepsilon-$greedy can be used to choose the action a_t from the action space $\mathcal{A}$ as

$$a_t = \begin{cases} \underset{a}{\text{argmax}}\{Q(O_t, \mathcal{A}, \theta_V)\}, & x < 1 - \varepsilon, \\ \text{a random action} \in \mathcal{A}, & \text{otherwise}, \end{cases} \tag{16}$$

where $\varepsilon \in [0, 1]$ defines the percentage of exploration, and $x \in [0, 1]$ is randomly chosen for the selection of actions. Thus, the $\varepsilon-$greedy method can achieve the balance between exploitation and exploration. After executing the action a_t, the state of the environment becomes O_{t+1} and a scalar reward r_t can be obtained by the agent. Then, the DQL agent will choose the action a_{t+1} and repeat the above procedure until one of the termination conditions such as the maximum number of steps N achieved.

The experience of each control step $E_t = \{O_t, a_t, r_t, O_{t+1}\}$ will be stored at experience replay memory $Me = \{E_1, E_2, ...E_m\}$ with size m. The goal of the DQL agent is to find Q^* the optimal Q-function that gives the maximum expected return at the end of each episode

$$Q^* = \underset{\theta_V}{\text{argmax}} \sum_{t=0}^{N-1} Q(O_t, a_t, \theta_V). \tag{17}$$

At each learning step, Mini-batch samples $Mb_{samples}$ with size K will be randomly selected from Me for training purpose. Then, the prediction value $Q(O_t, a_t, \theta_V)$ and the target-value $\max_a\{Q(O_{t+1}, a, \theta_T)\}$ for each sample of $Mb_{samples}$ will be computed. For supervised learning, the Mean Square Error (MSE) is often adopted to evaluate loss between predict and target Q-values

$$l = \text{MSE}(r_t + \gamma(\max_a\{Q(O_{t+1}, a, \theta_T)\}) - Q(O_t, a, \theta_V)), \tag{18}$$

where, γ is the discount reward parameter. At the end of the learning step, the Gradient Descent (GD) optimizer with learning rate α will be used to update the weights θ_V of the value network to minimize the loss value l as

$$\theta_{V+1} \leftarrow \theta_V + \alpha(\nabla_{\theta_V} l|_{\theta_V}), \tag{19}$$

where $\nabla_{\theta_V} l|_{\theta_V}$ is the gradient of loss with respect to θ_V. However, the weights θ_T are a delayed copy of the θ_V, and they will be updated as

$$\theta_T \leftarrow \theta_V, \tag{20}$$

every Z episodes to be equal to the weights of the value network θ_V. The learning procedure for DQL agent keeps repeating until any of the termination conditions, like the maximum number of episodes, is achieved. At the end of training, DQL agent is expected to converge to the optimal control policy.

The training and learning process of Dueling DQL is the same as DQL but with improved neural architecture. The model of a dueling neural network is created from two networks, the state value network $V(O)$ and the advantage network $A(O, a)$. Both networks combine the same input layer to receive the same input state O, and the same aggregation output layer to give the following state action value.

$$Q(O, a, \theta) = V(O) + A(O, a). \tag{21}$$

$V(O)$ is a value of state O, while $A(O, a)$ is the advantage of choosing each action $a \in \mathcal{A}$ of the action space in state O.

IV. Modified Dueling-DQL for quantum gate design

The quantum gate for $n-$ qubits, as explained in Section II, is represented by a unitary matrix $U \in \mathbb{C}^{2^n \text{x} 2^n}$ with complex elements. Based on MDP, the DQL agent needs a state O_t for each control step to choose the action a_t. In order to compute the state O_t for the unitary U_t, we perform the following transformation

$$U_t = \begin{bmatrix} U_{1,1} & ... & U_{1,2^n} \\ . & . & . \\ . & . & . \\ U_{2^n,1} & ... & U_{2^n,2^n} \end{bmatrix} \rightarrow O_t = \begin{bmatrix} \text{Re}(U_{1,1}) \\ \text{Im}(U_{1,1}) \\ . \\ . \\ \text{Re}(U_{2^n,2^n}) \\ \text{Im}(U_{2^n,2^n}) \end{bmatrix} \tag{22}$$

where state $O_t \in \mathbb{R}^{2(2^{2n})}$ is a vector of real parts $\text{Re}(U_{i,j})$ and imaginary parts $\text{Im}(U_{i,j})$ with $i, j \in \{1, 2.., 2^n\}$. In this

work, we introduce a few improvements for the DuDQL approach for the optimal and robust quantum gate design problems.

A. Delayed Reward Function and Modified Experience Memory

The RL agent based on MDP will receive a reward r_t after executing action a_t. The reward r_t defines the quality of the action applied to minimize the performance function. The RL agent aims to minimize the performance function by maximizing the accumulated rewards during each episode. For the quantum gate design problems described in Section II, the goal is to construct the proper control protocol to approximate the quantum state to target gate U_T. Thus, the reward function for the MDP-based RL agent is

$$r_t = \begin{cases} -log(\bar{J}), & t = T, \\ 0, & \text{otherwise}, \end{cases} \tag{23}$$

where the subscript t indicates the control step. That is, the reward will only be given to the RL agent for the last step of each episode, and zero reward for the rest of the control steps. Thus, the convergence can be sped up. We propose the semi-Markov method for the DQL agent to receive the same reward $(r_1, r_2, ...r_N)$ after each episode for all control steps to speed up convergence. Based on the reward function, the reward value r_t increase with decreasing in $\bar{J}$,

$$r_1 = r_2 = ..., r_N = -log(\bar{J}). \tag{24}$$

In addition, we keep track of the objective function value $\bar{J}$ for each episode during training and save the experience M^* with the lowest value. After a certain number of episodes, the best transition will be added to the memory of the experience. This modification in experience memory increases the likelihood that the best experience is used to train the value network. As a result, it may be able to assist the RL agent in obtaining a good policy.

B. Modified Action Selection Procedure

The RL agent would choose the action by trial and error using the $\varepsilon-$greedy in (16). The actions are chosen at random or based on the Q-values. We propose a modified $\varepsilon-$greedy to choose the actions as

Algorithm 1 Modified $\varepsilon-$Greedy Procedure

if $(t < N)$ **then** ▷ $\varepsilon-$Greedy

$$a_t = \begin{cases} \underset{a}{\text{argmax}}\{Q(O_t, \mathcal{A}, \theta_V)\}, & x < 1 - \varepsilon, \\ \text{a random action} \in \mathcal{A}, & \text{otherwise}, \end{cases}$$

else

$a_t = \underset{a}{\text{argmin}}\{J|_{a\in\mathcal{A}})\}$. ▷ Deterministic Process

end if

The modified $\varepsilon-$greedy action selection method is the same with the standard $\varepsilon-$greedy for the selection of actions at each time step, with the exception that the final control pulses will be chosen deterministically. This deterministic process is used at the last control step, which would speed up the RL agent's ability to identify the optimal results.

C. Soft Update Procedure for θ_T

Based on the DQL standard training procedure, the weights of the target network θ_T will be updated to be a copy of θ_V every Z episodes as in (20). The Q-values of the target network will be used as reference for training the DQL agent. Thus, the rate of updating θ_T will influence the converging speed of the DQL agent to the optimal Q-function. Rather than using the hard update for θ_T as in (20), we update the weights of the target network every Z episodes to the following function

$$\theta_T \leftarrow (1 - \tau)\theta_T + \tau\theta_V, \tag{25}$$

where $\tau \in [0, 1]$ is a weight parameter that defines the percentage of update for θ_T. This soft update procedure would help DQL agent to achieve a better convergence behavior to a better policy by avoiding catastrophic jumps for values of θ_T out of the optimal results.

Combining the semi-Markov modified DQL method with the modified $\varepsilon-$greedy technique and the modified experience memory, our ultimate algorithm for the optimal and robust quantum gate design is outlined in Algorithm 2.

Algorithm 2 MDuDQL for Robust Quantum Gate Design

Input: Initial unitary U_0, Target unitary, U_T, Evolution time T, Number of episodes N_e, Actions space $\mathcal{A}$, Control steps N, Uncertainty boundary $[\epsilon_l, \epsilon_h]$.

Distribution of uncertain parameter:
Construct a set with a uniform distribution $D = \{\epsilon_l, \epsilon_2, \epsilon_3, ..., \epsilon_{n-1}, \epsilon_h\}$

for j=1,2,....,N_e **do** ▷ Loop for episodes
  Initialize unitary operator to U_0.
  Randomly choose uncertainty values ϵ' from D.
  for t=1,2,....,N **do** ▷ Beginning of episode
    Choose the action a_t according to Algorithm 1.
    Compute $U_t = e^{-iH(a_t,\epsilon')dt}U_{t-1}$
    Save the tuple $E_t = \{U_{t-1}, a_t, U_t\}$
    if $mod(t, z) == 0$ and $j > 1$ **then**
      Updates parameters θ_V as in (19).
    end if
  end for ▷ End of Episode
  Test control protocol $\{a_1, a_2, ..., a_N\}$ for all uncertain values in vector D.
  Compute final unitary: $\{U_{f_1}, U_{f_2}, ..., U_{f_n}\}$.
  Compute the objective function $\bar{J}$ using (13).
  Reward $r = -log(\bar{J})$.
  Store $M_j = \{\{E_1, r\}, \{E_2, r\}, ..., \{E_{N_c}, r\}\}$ in the replay experiences memory M_e.
  if $mod(j, Z) == 0$ and $j > 1$ **then**
    Updates parameters θ_T as in (25).
  end if
  Add best discovered experience M^* to M_e.
end for

V. Results and Discussion

In this section, we demonstrate simulation results for the optimal and robust quantum gate design problems described in Section II. The proposed Modified Dueling DQL (MDuDQL) method is employed to generate three quantum

gates: H, T and S-gate. The results of MDuDQL are presented and compared with the dueling deep Q-learning algorithms based on both the Markov and the semi-Markov process which are referred as DuDQL$_1$, DuDQL$_2$, DuDQL$_3$, and DuDQL$_{\text{nR}}$. Specifically, DuDQL$_1$, DuDQL$_2$, and DuDQL$_3$ are standard DuDQL with different hyperparameters, whilst, DuDQL$_{nR}$ is the standard DuDQL combined with proposed n-step delay function. The results of the robust gate design have been generated on an National Computational Infrastructure (NCI) high-performance computer, by using 48 CPUs and 8 Gbit's memory for each problem, and the hyperparameters used for the proposed results are presented in Table I.

TABLE I
PARAMETER VALUES OF VARIOUS ALGORITHMS

Parameter	**Value**
Learning Rate (α)	0.005 , DuDQL$_3$ = 0.001
Reward Discount (γ)	0.95 , DuDQL$_2$ = 0.99
Number of Episodes (E)	30000
Size of Hidden-Layer	512
Experience Memory Size (m)	25000
Size of Mini-batch (K)	64
Training Predict Weights (n)	19 (Steps)
Replacement Target Weights (Z)	30 (Episodes)
Epsilon Updating Step ϵ_{step}	0.0001
Control Steps (N)	38
Soft Update Percentage (τ)	0.01

A. Results for optimal quantum gate design

This section contains the results of the proposed approach for the optimal design of the quantum gate of a single-qubit system. The performance of the proposed approach for optimal gate design has been compared with classical Dueling DQL methods. The goal of the optimal gate design is constructing the quantum gate with minimum infidelity explained in (8). The numerical results of the best achieved infidelity for 5 trials of MDuDQL, DuDQL$_{\text{nR}}$, DuDQL$_1$, DuDQL$_2$ and DuDQL$_3$ are shown in Figure 1.

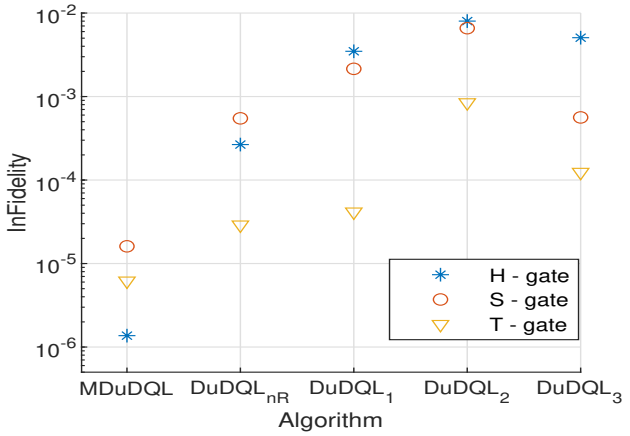

Fig. 1. The best achieved value of the cost function for quantum gates by using MDuDQL, DuDQL$_{\text{nR}}$, DuDQL$_1$, DuDQL$_2$ and DuDQL$_3$

As shown in Figure 1, the proposed MDuDQL has succeeded in constructing the quantum gates with the lowest infidelities around 10^{-5}, 10^{-5} and 10^{-6} for the H, T and S-gate correspondingly. The infidelities of the other four algorithms are more than 10 times higher than the MDuDQL, and DuDQL$_2$ gives the worst performance.

The average achieved infidelity of S-gate design vs the number of episodes for the DuDQL algorithms is shown in Figure 2. The MDuDQL as shown in Figure 2, has exhibited the best training progress among the five algorithms for determining the lowest infidelity solution. Although, the MDuDQL provides similar performance with the DuDQL$_{nR}$ for episodes under 800, it still outperforms other DuDQL algorithms. On the other hand, the traditional DuDQL algorithms with different hyper-parameters, have been stuck into the local optimal solutions and have failed to converge to a better policy. The n-delay reward function, as shown for the results of DuDQL$_{nR}$, has increased the agent stability throughout training, which allows it to stay close to the best found outcomes. In addition, the modified action selection mechanism has helped the MDuDQL agent to skip out of local optimal results. Also, the enhanced experience replay approach and soft update function for the target network have improved MDuDQL agent training to converge to global or near-global optimal results.

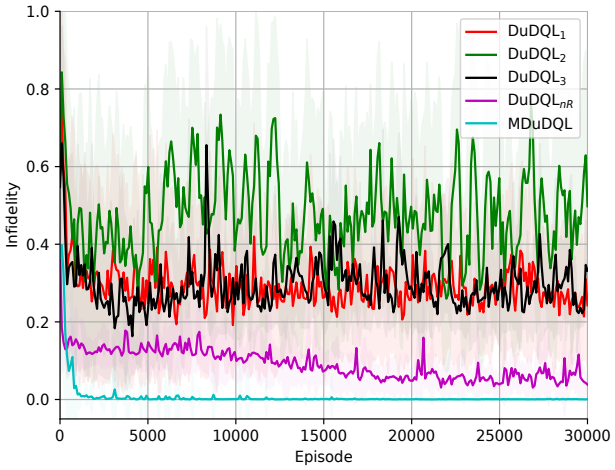

Fig. 2. Average value of the performance function for different DuDQL algorithms for the optimal gate design of the S-gate.

When control resources are limited, the traditional DuDQL may struggle to find high-fidelity control protocols. The proposed technique improves DuDQL performance in overcoming the exploration-exploitation balancing problem, discovering the global optimal solutions, and avoiding getting stuck in local optimal outcomes, as illustrated in the results above.

B. Robust gate design

In this section, the results of the suggested method for the robust gate design are compared to the classical DuDQL algorithms. We sample the uncertainty range to a uniform distribution consisting of 441 samples for training purposes for the problem described in Section II-B. The goal is to construct a robust control protocol that can improve the worst case while increasing the average fidelity, as described in (13). In (13), the value of the weight parameter w has been set to $w = 0.7$. Table II shows the numerical results of the

fidelity gained on average and the worst case of the best results acquired on different DuDQL.

TABLE II
AVERAGE AND WORST ACHIEVED FIDELITY FOR DIFFERENT DUDQL ALGORITHMS FOR TRAINING SET

Algorithms	H-Gate		S-Gate		T-Gate	
	Avg	Min	Avg	Min	Avg	Min
MDuDQL	0.994	0.984	0.997	0.982	0.994	0.990
DuDQL_{nR}	0.994	0.980	0.996	0.982	0.996	0.985
DuDQL_1	0.991	0.967	0.995	0.975	0.979	0.986
DuDQL_2	0.991	0.969	0.996	0.976	0.979	0.943
DuDQL_3	0.993	0.976	0.994	0.971	0.988	0.964

As shown in Table II, the discovered control protocols for H, and S-gate by the MDuDQL, DuDQL_{nR} and standard DuDQL have similar average fidelity. However, as compared to standard DuDQL control protocols, the MDuDQL, DuDQL_{nR} control protocols are better in terms of improving the minimum fidelity. In addition, when compared to standard DuDQL, the results of the T-gate show that MDuDQL and DuDQL_{nR} perform better in identifying the robust control protocol on a training set of uncertainty. However, Table II contains the best findings discovered for the training set which is not enough to determine whether the discovered control protocols are robust or overestimated for the training set of uncertainty. For the robustness test, we use 10201 new sets of uncertainty to evaluate each algorithm. The percentage of sets with high fidelity ≥ 0.99 is shown in Figure 3 .

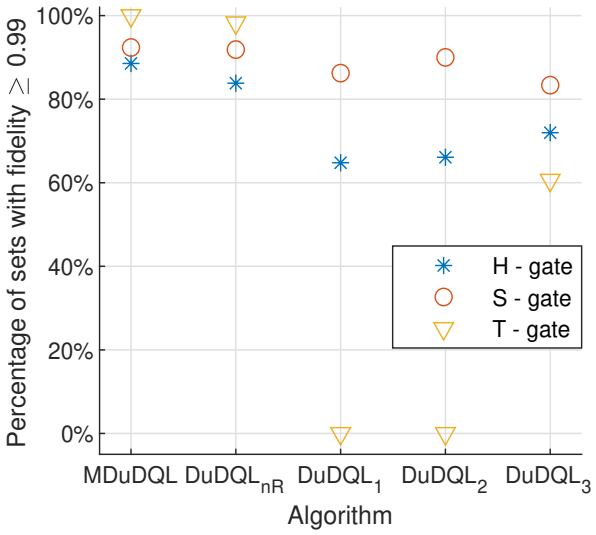

Fig. 3. Percentage of fidelity best achieved ≥ 0.99 of the robust control protocol for the testing set among a total number of 10201 sets.

The control protocols discovered by the MDuDQL are successful in obtaining high fidelity of more than 0.99 for new sets of uncertainty for roughly 90%, 95%, and 100% for the H, S and T-gate correspondingly. Similar results for the DuDQL_{nR} like the MDuDQL for the different quantum gates. The performance of the discovered control protocols by standard DuDQL compared to the MDuDQL and DuDQL_{nR} are the worst, especially for the T-gate. In conclusion, the control protocols found by MDuDQL DuDQL_{nR} with proposed n-delay reward function, are the most robust to achieve high fidelity ≥ 0.99 for most of the new uncertainty values within the specified range compared to the standard DuDQL.

VI. CONCLUSION

The optimal and robust design of quantum gates is essential for quantum computation and communication. In this work, we considered the optimal and robust quantum gate design of a single quantum system with limited control resources and uncertain parameters, aiming to generate a quantum gate with high fidelity. We proposed the MDuDQL algorithm, which is a novel modified version of the DuDQL. The modified action selection procedure, soft update procedure and improved replay memory were proposed. Numerical results show that the modified dueling DQL is effective for designing the optimal and robust quantum gates of the single qubit system. The modified algorithm shows an improved robustness and the ability to converge to the best control policy. The proposed modifications also have the potential to be integrated in more advanced free model Q-learning methods. We will keep exploring the possibility of applying this novel approach for the optimal and robust quantum gate design of multi-qubit systems.

REFERENCES

[1] A. W. Harrow, and A. Montanaro, "Quantum computational supremacy," *Nature,* vol. 549, no. 7671, pp.203-209, 2017.

[2] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, and M. Broughton, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *Nature Physics,* vol. 17, no. 3, pp.332-336 2021.

[3] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum,* vol. 2, p. 79, 2018.

[4] K. S. Chou, J. Z. Blumoff, C. S. Wang, P. C. Reinhold, C. J. Axline, Y. Y. Gao, L. Frunzio, M. H. Devoret, L. Jiang, and R. J. Schoelkopf, "Deterministic teleportation of a quantum gate between two logical qubits," *Nature,* vol. 561, no. 7723, pp.368-373, 2018.

[5] D. Awschalom, K. K. Berggren, H. Bernien, S. Bhave, L. D. Carr, P. Davids, S. E. Economou, D. Englund, A. Faraon, M. Fejer, and S. Guha, "Development of quantum interconnects (quics) for next-generation information technologies," *PRX Quantum,* vol. 2, no. 1, p.017002, 2021.

[6] D. Dong, and I. R. Petersen, "Quantum control theory and applications: a survey," *IET Control Theory & Applications,* vol. 4, no. 12, pp. 2651-2671, 2010.

[7] A. Koswara, V. Bhutoria, and R. Chakrabarti, "Quantum robust control theory for Hamiltonian and control field uncertainty," *New Journal of Physics,* vol. 23, no. 6, p.063046, 2021.

[8] R. L. Kosut, M. D. Grace, and C. Brif, "Robust control of quantum gates via sequential convex programming," *Physical Review A,* vol. 88, no. 5, p.052326, 2013.

[9] B. Riaz, C. Shuang, and S. Qamar, "Optimal control methods for quantum gate preparation: a comparative study," *Quantum Information Processing,* vol. 18, no. 4, pp.1-26, 2019.

[10] F. K. Wilhelm, S. Kirchhoff, S. Machnes, N. Wittler, and D. Sugny, "An introduction into optimal control for quantum technologies," *arXiv preprint arXiv:2003.10132,* 2020.

[11] H. Ma, and C. Chen, "Several developments in learning control of quantum systems," *IEEE International Conference on Systems, Man, and Cybernetics,* pp. 4165-4172, 2020.

[12] B. Andresen, K. H. Hoffmann, J. Nulton, A. Tsirlin, and P. Salamon, "Optimal control of the parametric oscillator," *European journal of physics,* vol. 32, no. 3, p.827, 2011.

[13] D. Dong, C. C. Shu, J. Chen, X. Xing, H. Ma, Y. Guo, and H. Rabitz, "Learning control of quantum systems using frequency-domain optimization algorithms," *IEEE Transactions on Control Systems Technology,* vol. 29, no. 4, pp.1791-1798, 2021.

[14] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, and D. Sugny, "Training Schrödinger's cat: quantum optimal control," *The European Physical Journal D,* vol. 69, no. 12, pp.1-24, 2015.

[15] F. Dolde, V. Bergholm, Y. Wang, I. Jakobi, B. Naydenov, S. Pezzagna, J. Meijer, F. Jelezko, P. Neumann, T. Schulte-Herbrüggen, and J. Biamonte, "High-fidelity spin entanglement using optimal control," *Nature communications,* vol. 5, no. 1, pp.1-9, 2014.

[16] G. Jäger, D. Reich, M. H. Goerz, C. P. Koch, and U. Hohenester, "Optimal quantum control of bose-einstein condensates in magnetic microtraps: Comparison of grape and krotov optimization schemes," *arXiv preprint arXiv:1409.2976,* 2014.

[17] M. H. Goerz, D. M. Reich, and C. P. Koch, "Optimal control theory for a unitary operation under dissipative evolution," *New Journal of Physics,* vol. 16, no. 5, p.055012, 2014.

[18] N. A. Petersson, and F. Garcia, "Optimal control of closed quantum systems via b-splines with carrier waves," arXiv preprint arXiv:2106.14310, 2021.

[19] S. Günther, N. A. Petersson, and J. L. DuBois, "Quantum optimal control for pure-state preparation using one initial state," *AVS Quantum Science,* vol. 3, no. 4, p.043801, 2021.

[20] M. H. Goerz, D. Basilewitsch, F. Gago-Encinas, M. G. Krauss, K. P. Horn, D. M. Reich, and C. P. Koch, "Krotov: A Python implementation of Krotov's method for quantum optimal control", *SciPost Phys*, vol. 7, no. 6, p. 80, 2019.

[21] D. Dong, and I. R. Petersen, "Quantum estimation, control and learning: opportunities and challenges," *Annual Reviews in Control, in press, DOI: 10.1016/j.arcontrol.2022.04.011,* 2022.

[22] D. Daems, A. Ruschhaupt, D. Sugny, and S. Guerin, "Robust quantum control by a single-shot shaped pulse," *Physical Review Letters,* vol. 111, no. 5, p.050404, 2013.

[23] D. Dong, and Y. Wang, "Several recent developments in estimation and robust control of quantum systems," *Australian and New Zealand Control Conference,* pp. 190-195, 2017.

[24] D. Dong, X. Xing, H. Ma, C. Chen, Z. Liu, and H. Rabitz, "Learning-based quantum robust control: Algorithm, applications and experiments," *IEEE Transactions on Cybernetics,* vol. 50, no. 8, pp. 3581-3593, 2020.

[25] Y.X. Zeng, J. Shen, S.C. Hou, T. Gebremariam and C. Li, "Quantum control based on machine learning in an open quantum system, " *Physics Letters A,* vol. 384, no. 35, p. 126886, 2020.

[26] R. B. Wu, B. Chu, D. H. Owens, and H. Rabitz, "Data-driven gradient algorithm for high-precision quantum control," *Physical Review A,* vol. 97, no. 4, p.042122, 2018.

[27] R.B. Wu, H. Ding, D. Dong and X. Wang, "Learning robust and high-precision quantum controls," *Physical Review A,* vol. 99, no. 4, p. 042327, 2019.

[28] E. Zahedinejad, J. Ghosh, and B. C. Sanders, "High-fidelity single-shot toffoli gate via quantum control," *Physical Review Letters,* vol. 114, no. 20, p. 200502, 2015.

[29] D. Dong, C. Wu, C. Chen, B. Qi, I. R. Petersen and F. Nori, "Learning robust pulses for generating universal quantum gates," *Scientific Reports*, vol. 6, p. 36090, 2016.

[30] C. Wu, B. Qi, C. Chen, and D. Dong, "Robust learning control design for quantum unitary transformations," *IEEE Transactions on Cybernetics*, vo. 47, pp. 4405-4417, 2017.

[31] E. Zahedinejad, J. Ghosh, and B. C. Sanders, "Desgining high-fidelity single-shot three-qubit gates: A machine-learning approach," *Physics Review Applied,* vol. 6, no. 5, p. 054005, 2016.

[32] R. S. Sutton, and A. G. Barto, *Reinforcement learning: An introduction,* 2nd edition, MIT Press, 2018.

[33] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *arXiv preprint arXiv:1811.12560,* 2018.

[34] G. Lample, and D. S. Chaplot, "Playing FPS games with deep reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 31, no. 1, 2017.

[35] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, and Y. Chen, "Mastering the game of go without human knowledge," *Nature,* vol. 550, no. 7676, pp. 354-359, 2017.

[36] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, and T. Lillicrap, "Mastering atari, go, chess and shogi by planning with a learned model," "*Nature,* vol. 588, no. 7839, pp.604-609, 2020.

[37] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, and C. Donner, "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature,* vol. 602, no. 7897, pp.414-419, 2022.

[38] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, "Universal quantum control through deep reinforcement learning," *npj Quantum Information,* vol. 5, no. 1, p. 33, 2019.

[39] H. Ma, D.Dong, S.X. Ding, and C. Chen, "Curriculum-based deep reinforcement learning for quantum control," *IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2022.3153502*, 2020.

[40] T. Haug, R. Dumke, L.C. Kwek, C. Miniatura, and L. Amico, "Machine-learning engineering of quantum currents," *Physical Review Research,* vol. 3, no. 1, p. 013034, 2021.

[41] Z. An, and D.L. Zhou, "Deep reinforcement learning for quantum gate control," *EPL Europhysics Letters,* vol. 126, no. 6, p.60002, 2019.

[42] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, "Reinforcement learning with neural networks for quantum feedback, *Physical Review X,* vol. 8, no. 3, p. 031084, 2018.

[43] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, "Optimizing quantum error correction codes with reinforcement learning," *Quantum,* vol. 3, p.215, 2019.

[44] H. Xu, J. Li, L.Liu, Y. Wang, H. Yuan, and X. Wang, "Generalizable control for quantum parameter estimation through reinforcement learning," *npj Quantum Information,* vol. 5, no. 1, pp. 1-8, 2019.

[45] P. Peng, X. Huang, C. Yin, L. Joseph, C. Ramanathan, and P. Cappellaro, "Deep reinforcement learning for quantum Hamiltonian engineering," *arXiv preprint arXiv:2102.13161,* 2021.

[46] J. Qian, R. Fruit, M. Pirotta, and A. Lazaric, "Exploration bonus for regret minimization in undiscounted discrete and continuous Markov decision processes," *arXiv preprint arXiv:1812.04363,* 2018.

[47] M. H. Goerz, D. M .Reich, and C. P. Koch, "Optimal control theory for a unitary operation under dissipative evolution," *New Journal of Physics,* vol. 16, no. 5, p.055012, 2014.

[48] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine,* vol. 34, no. 6, pp. 26-38, 2017.

[49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 7540, pp. 529-533, 2015.