2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
17-20 October, 2021. Melbourne, Australia

A Modified Deep Q-Learning Algorithm for Control of Two-qubit Systems

Omar Shindi, Qi Yu, Daoyi Dong*†

2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-4207-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMC52423.2021.9658732

***Abstract*— Quantum control refers to the manipulation of dynamical quantum systems to force them to complete given tasks such as preparing a desired state and tracking a designed trajectory. We consider the state preparation problem of a two-qubit closed quantum system from an initial to a desired state. The aim is to achieve a high fidelity in a given fixed time with limited control resources. The Deep Q-learning (DQL) for solving the quantum state preparation problem is explored in this paper. We propose a novel semi-Markov DQL algorithm based on a modified action selection procedure and improved replay memory to enhance performance of standard DQL algorithm. The proposed algorithm shows high performance for discovering high-fidelity control protocols and for converging to a good policy, compared with standard DQL. The proposed modifications enhance the exploration-exploitation ability for DQL agent and the robustness for solving quantum control problem with high-fidelity at different numbers of control steps. Numerical results on a two-qubit closed system show effectiveness of the proposed algorithm.**

I. Introduction

Effective quantum state manipulation is considered essential for applications of quantum control in different fields, like quantum optics [1] and quantum computing [2], [3]. Whereas, quantum technologies are considered to be promising to offer a high computation capability [4], [5] and secure telecommunication system [6]. Recently, quantum control field has witnessed considerable developments [7], [8], where a basic task is to develop various estimation and control strategies to achieve state manipulation of different quantum systems such as atoms, molecules, electrons and photons [9], [10]. In general, a quantum state manipulation problem can be formulated as an optimization problem, aiming to find proper control pulses that minimize a cost function to achieve a specific task such as steering a quantum system to a desired state. Classical control methods, like Lyapunov control [11], optimal control theory [12], have been proposed for solving quantum state control problems. Learning algorithms based on gradient [13] and differential evolution [14], [15] have also been proposed for robust quantum control and high-fidelity quantum gate design.

Recently, machine learning algorithms have attracted attention in the quantum domain [16], [17] and have been employed in quantum control problems [18], [19]. For example, supervised deep learning has been employed for the robust control of quantum systems [20]. Reinforcement learning (RL) techniques [21], [22] have shown high potential for solving different kinds of quantum physics problems even without known accurate model [24]-[35]. Inherent advantage of RL over other machine learning methods is the ability to learn by trial and error through interacting with environment, and the ability to balance between exploitation of existing knowledge and exploration of unknown parts of the control landscape [23].

Classical RL algorithms based on Q-learning have been applied successfully for solving quantum control problems [24], [26], like quantum state preparation in glassy phase where optimal state manipulation is very hard [24], [25]. Deep Reinforcement Learning (DRL) after showing a good performance for solving realistic complex problems [27], has attracted researchers for solving challenging quantum control problems [29] like reducing entropy production in closed quantum systems [28]. DRL algorithms have shown a generalized ability on similar scenarios for quantum gate design [30], [31] and quantum state preparation problems [32]. Moreover, the DRL algorithms were employed for parameter estimation problem [33], [34] and they have shown outstanding performance for discovering quantum error correction strategies [35].

Finding a high-fidelity control protocol still remains a challenge, specially with limited control resources. In this paper, we propose a novel semi-Markov Modified DQL (MDQL) algorithm based on modified action selection procedure and improved replay-memory for solving a control problem of a two-qubit system. The MDQL method is then compared with the classical DQL based Markov and semi-Markov procedures. Numerical results show a superior performance of MDQL for reaching global optimal results which achieve a higher fidelity compared with the standard DQL.

The remainder of this paper is organized as follows. Section II illustrates the quantum control problem. Section III presents the DQL process and proposes a modified action selection procedure. The improvement on replay memory and semi-Markov procedure is analysed and explained. The numerical results and discussion are provided in Section IV. Concluding remarks are given in Section V.

II. Quantum System

Dynamics of a closed quantum system can be represented using the Schrödinger equation

$$i\hbar|\dot{\psi}(t)\rangle = H|\psi(t)\rangle, \tag{1}$$

where $|\psi(t)\rangle$ is the quantum state at time t, and $\hbar$ is the Plank constant (assuming $\hbar = 1$), while i is unit imaginary

*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.

† School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (email: omar.shindi.ca@gmail.com, yuqivicky92@gmail.com, daoyidong@gmail.com).

number. The Hamiltonian H can be written as

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

where H_0 is the free Hamiltonian and H_c is control Hamiltonian that can be fabricated. The state of a closed B-qubit system can be represented by a state vector of size 2^B in a complex Hilbert space $\mathcal{H}$. In particular, the state vector of a two-qubit system can be formed as

$$|\psi\rangle = |\psi\rangle^1 \otimes |\psi\rangle^2, \tag{3}$$

where $|\psi\rangle^1$ and $|\psi\rangle^2$ represent the states of the first and the second qubits, respectively, and the operation $\otimes$ denotes tensor product. For current work, the Hamiltonian of a two-qubit system is considered as

$$H_0 = (S_z \otimes \mathbb{I})(\mathbb{I} \otimes S_z) + (S_y \otimes \mathbb{I} + \mathbb{I} \otimes S_y), \tag{4}$$

$$H_c = u_1(S_x \otimes \mathbb{I}) + u_2(\mathbb{I} \otimes S_x), \tag{5}$$

where u_1, u_2 are external control fields of the first and second qubits, respectively, in x direction, while S_x, S_y, S_z are spin operators. Specifically, we have

$$S_x = 0.5\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, S_y = 0.5\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}, S_z = 0.5\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \tag{6}$$

and $\mathbb{I}$ is the identity matrix of size 2×2

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

If we assume $|\psi_j\rangle$ is current quantum state at t_j for Hamiltonian H, the next quantum state $|\psi_{j+1}\rangle$ at t_{j+1} can be found by using following time-evolution equations

$$|\psi_{j+1}\rangle = U_j |\psi_j\rangle, \tag{8}$$

$$U_j = e^{(-iH(u_j)dt)}, \tag{9}$$

where U_j is time evolution operator for applied control field u_j during $[t_j, t_{j+1}]$, and $dt = t_{j+1} - t_j$ is time duration for the control pulse u_j. For a fixed total evolution time T and the specified number of time steps N, we assume that, the time duration is equal $dt = T/N$.

The objective of quantum state preparation is finding a proper control protocol to derive the quantum system from the initial state $|\psi_0\rangle = |01\rangle$ to the target state $|\psi_T\rangle = |10\rangle$ at fixed total evolution time $T \in [0, 4]$, with high fidelity, while there are constraints on the control signal $u_1, u_2 \in \{2, 0, -2\}$. The fidelity between the achieved final quantum state $|\psi_f\rangle$ and the desired target state $|\psi_T\rangle$ can be measured as

$$F = |\langle\psi_f|\psi_T\rangle|^2, \tag{10}$$

where $\langle\psi_f|$ is the conjugate and transpose of $|\psi_f\rangle$. This work aims to find a proper control sequence of u to achieve the control task with high fidelity.

III. Algorithm Description

A. Deep Q-learning Algorithm

Deep Q-learning (DQL) is a value-based RL algorithm using Neural Networks (NNs) to approximate Q-values for action-state pairs as a replacement for tabular representation. Thus, the DQL method is able to solve more complex or high-dimensional problems arising like video games [36], [37].

Basically, DQL agent contains two neural networks of the same architecture: the value-network with weights θ_V, and target-network with weights θ_T. The value-network receives current state s_j and returns Q-values $Q(s_j, a, \theta_V)$ for all allowed actions $a \in [a_1, a_2..., a_p]$. The target-network receives next-state s_{j+1} and returns Q-values $Q(s_{j+1}, a, \theta_T)$ for all actions. At instant time j and current state s_j, DQL agent chooses an action a_j based on a specified procedure, like epsilon-greedy method, to interact with environment and moves to next state s_{j+1},

$$a_j = \begin{cases}\underset{a}{\text{argmax}}\{Q(s_q, a, \theta_V)\}, & x < 1 - \epsilon,\\ \text{a random action} \in A, & \text{otherwise},\end{cases} \tag{11}$$

where $\epsilon \in [0, 1]$ is epsilon-greedy parameter, and x is choosing randomly in $[0, 1]$ to achieve balance between exploitation and exploration for action selection from action space $A = [a_1, a_2, ..., a_p]$. Then, DQL agent will receive a reward r_j for current state-transition of applied action a_j and based on a specified rewarding function that defines quality of action selection at different states to achieve control objectives. The DQL agent aims to find the best control sequence that is giving the highest accumulating rewards $R = \sum_{j=1}^{N} r_j$. State-transition experience $E_j = (s_j, a_j, r_j, s_{j+1})$ will be stored at experience memory $Me = \{E_1, E_2, ...E_m\}$ with size m for later use of selecting randomly a training Mini-batch samples $Mb_{samples}$ with size K to train the value network.

The target-network is required for supervised learning to compute target-value or expected maximum Q-value $\max_a\{Q(s_{j+1}, a, \theta_T)\}$ at next-state s_{j+1} for each sample of $Mb_{samples}$ by applying following Q-learning update with a discount reward γ

$$Q_T = r_j + \gamma(\max_a\{Q(s_{j+1}, a, \theta_T)\}), \tag{12}$$

then, the Mean Square Error (MSE) is adopted to evaluates loss between predict and target Q-values

$$l = MSE(Q(s_j, a, \theta_V) - Q_T). \tag{13}$$

Parameter θ_V of the value-network will be updated to minimise the loss value l by using a Gradient Descent (GD) optimizer with learning rate α

$$\theta_{V+1} \leftarrow \theta_V - \alpha(\nabla_{\theta_V} l|_{\theta_V}), \tag{14}$$

where $\nabla_{\theta_V} l|_{\theta_V}$ is the gradient of loss with respect to θ_V. However, weights of the target-network will be updated as $\theta_T \rightarrow \theta_V$ 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.

B. Rewards and Modified Experience Memory

For Markov process, the RL agent will receive a reward r_j which defines quality of state-action pair (s_j, a_j) after each state transition $s_j \xrightarrow{a_j} s_{j+1}$. While the ultimate goal of RL agent is to maximize the collecting reward R. For the considered quantum control problem described in Section II, the goal is to find a proper control sequence that steering a quantum system from the initial state s_0 to the desired target state s_T. Thus, the reward function for RL agent based on Markov process is formulated as

$$r_i = \begin{cases} 10 * F_f, & \text{final step,} \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

to give RL agent a reward only at the last step based on final fidelity F_f of each episode

$$F_f = |\langle s_f | s_T \rangle|^2. \tag{16}$$

where s_f is episode final state. Rather than giving rewards directly after each step and to speed up convergence, we propose the semi-Markov process for DQL agent to receive rewards $(r_1, r_2, ...r_N)$ after each episode as follows

$$r_1 = r_2 = ...r_N = 10 * F_{final}. \tag{17}$$

During training, we keep monitoring achieved final fidelity for each episode and store the transitions of the one with the highest achieved fidelity. The best transition will be added into the experience memory periodically after a specified number of episodes. This modification on experience memory can increase the chance of using the best experience for training the value network. As a result, it can assist RL agent to converge to a good policy.

C. Modified Action Selection Procedure

The value of ϵ as shown in (11) defines the percentage of exploitation and exploration, to prevent from sticking into local optimal results while assisting the RL agent to approach the global optimal results. A new composed procedure consisting of dynamic greedy method and normal decay function described in Algorithm 1 is adopted for our work. The dynamic greedy procedure keeps updating the value of ϵ from 0 to 1 and then backward from 1 to 0, as a searching method to find the best result. Then, the normal decay procedure ending with full exploitation will be applied to lead the RL agent to convergence.

Algorithm 1 Modified Dynamic Greedy Procedure

```
if (ε_switch == 1) then                          ▷ Dynamic or Normal Decay
    if (ε > 0 & S_e == 1) then
        ε = ε − ε_step                           ▷ Increase Exploitation
    else
        ε = ε + ε_step                           ▷ Increase Exploration
        if (ε ≥ 1) then                          } Dynamic Greedy
            S_e = 1
        else
            S_e = 0
        end if
    end if
else
    ε = ε − ε_step                               ▷ Normal Decay method
end if
```

In Algorithm 1, the value of ϵ_{switch} defines which method is used to update ϵ by value ϵ_{step}. We further improve the dynamic greedy action selection by applying a deterministic process at the last control step N to select action $a_N \in A$ that gives the highest fidelity for last state $F(s_{final})$ after trying all the allowed actions

$$a_N = \underset{a}{\operatorname{argmax}}\{F(s_{final}|_{a \in A})\}. \tag{18}$$

This extra step will be applied only when the dynamic greedy method is used to increase the speed and chance for RL agent to find the best results before going to greedy normal decay process. The proposed semi-Markov Modified DQL algorithm (MDQL) combined with the modified dynamic greedy procedure and the modified experience memory is described in Algorithm 2.

IV. Results and Discussion

In this section, we demonstrate simulation results for the control problem of the two-qubit system described in Section II. The results of the proposed MDQL method are given. For comparison, the classical deep Q-learning algorithms based on both the Markov and the semi-Markov process are also employed and the corresponding algorithms are referred as $\text{DQL}_1, \text{DQL}_2$. Used hyper parameters for proposed results are presented in Table I.

TABLE I
PARAMETER VALUES OF VARIOUS ALGORITHMS

Parameter	Value
Learning Rate (α)	0.005
Reward Discount (γ)	0.99
Number of Episodes (E)	10^5
Size of Hidden-Layer	125
Experience Memory Size (m)	20000
Size of Mini-batch (K)	64
Training Predict Weights (n)	2 (Episodes)
Replacement Target Weights (Z)	10 (Episodes)
Epsilon Updating Step ϵ_{step}	0.001
Dynamic Greedy Episodes z	10^4 (Episodes)
Control Steps (N)	5, 10, 20, 30

Algorithm 2 Modified DQL (MDQL) Algorithm

Input: Initial State s_0, Target State s_T, Discount Factor γ, Number of Episodes D, Actions Space A, Final Time T, Control steps N, Learning Rate α, Mini-Batch Size K, Experience Memory Size m, Replacement Target Weights Z, Dynamic Greedy Episodes z, Training Predict Weights n.

Initialization: Initialize θ_V and θ_T, $\epsilon_{switch} = 1$, $\epsilon = 1$

for E=1:D **do** ▷ Beginning of learning episodes
  $j = 1$, ▷ Control steps counter
  $s_j = s_0$, ▷ Initialize state
  M_{temp}=[], ▷ Initialize temp-memory
  while ($j \leq N$ & $F < 0.999$) **do**
    Generate a random variable $x \in [0, 1]$
    if ($E < z$) **then** ▷ Modified Dynamic Greedy
      if ($j < N$) **then**
        Choose action a_j according to Eq.(11)
      else ▷ At last time step $j = N$
        Choose action achieving highest Fidelity:
        $a_j = \underset{a}{\operatorname{argmax}}\{F(s_{j+1}|_{a \in A})\}$
      end if
    else ▷ Greedy Normal Decay
      Choose action a_j according to Eq.(11)
      $\epsilon_{switch} = 0$
    end if
    Take action a_j, observe s_{j+1}, r_j and fidelity F.
    Store experience $e_j = (s_j, a_j, r_j, s_{j+1})$ at M_{temp}
    $j = j + 1$
  end while
  Update all rewards in M_{temp} to final reward:
  $r_1 = r_2 = r_3 = ... = r_N$ ▷ semi-Markov process
  if ($F > F_b$) **then** ▷ Store Best Experience
    $M_b = M_{temp}$
    $F_b = F$
  end if
  Store M_{temp} to experience memory M
  if $mod(E, n) == 0$ **then** ▷ Every n Episodes
    Add M_b into M
    Randomly sample Mini-batch of size K from M:
    Compute $Q(s, a, \theta_V)$ for each sample
    Compute Q_T for each sample, according to Eq.(12)
    Compute loss according to Eq.(13)
    Update θ_V according to Eq.(14)
  end if
  Update θ_T every Z episodes
  Update ϵ based on Algorithm 1
end for

Numerical results of average achieved fidelity for 10 trials at different control steps for trained DQL_1, DQL_2 and MDQL are given in Table II. As shown, both the DQL_1 and DQL_2 fail for discovering high-fidelity control protocols, while the MDQL has succeeded for finding control sequences achieving high fidelity at different numbers of control pulses. The reason for the failure of the classical DQL may be due to the limited control resources which offer few high-fidelity solutions. Even though RL agent has succeeded to discover them during training, they appear rarely in replay memory, so the chance for using them to train RL agent is rather small. Instead, the proposed modified memory procedure can assist the RL agent in MDQL to overcome this problem by keeping the best discovered results all the time in reply memory experience to increase chance of using them for the training RL agent.

TABLE II
AVERAGED ACHIEVED FIDELITY BY TRAINED DQL AGENT ON 10 TRIALS AT DIFFERENT NUMBERS OF CONTROL PULSES

	Algorithms		
N	DQL_1	DQL_2	MDQL
5	0.86388	0.86388	0.99661
10	0.16219	0.86388	0.99729
20	0.68494	0.86388	0.99832
30	0.68494	0.86388	0.99887

When resources are limited and there are different constraints, high-fidelity control protocols may be difficult to be found in many applications. This situation is increasing challenge for achieving exploration-exploitation balance, to help DQL agent to discover global optimal solutions and prevent stuck into local optimal results. For example, Fig. 1 shows achieved fidelity for all control protocols that can be obtained for u_1 and u_2 at $N = 5$ control steps. As shown in Fig. 1, there is only one control protocol which can achieve fidelity higher than 0.99 among 59049 protocols. Hence, finding this protocol within a high density of local optimal solutions is a challenge for DQL agent, and it requires more advanced exploration-exploitation technique to discover a high-fidelity control protocol. As given in Table II, the DQL_1 and the DQL_2 agents easily get stuck to local optimal solutions. Whereas, the MDQL algorithm makes improvements on the action selection procedure. Thus, it has the potential to successfully reach global optimal results.

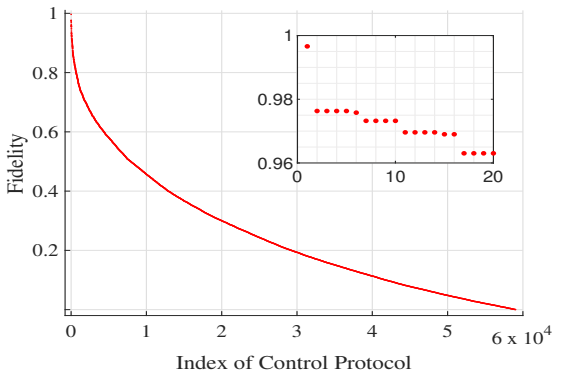

Fig. 1. Final fidelity of all control protocols at N=5

Numerical results of the instantaneous achieved fidelity at each episode for DQL_1 and DQL_2 for $N = 20$ control

pulses are presented in Fig. 2 and Fig. 3, respectively. The results show that both the DQL_1 and the DQL_2 are stuck into local optimal solutions with the final fidelity to be around 0.7 and 0.85, respectively, which are much lower than expected. It can also be seen that DQL_2 performs better than DQL_1 by achieving higher fidelity and converging faster due to the applying of the semi-Markov process. DQL_2 also shows stability for discovering the same results at different numbers of control steps as shown in Table II.

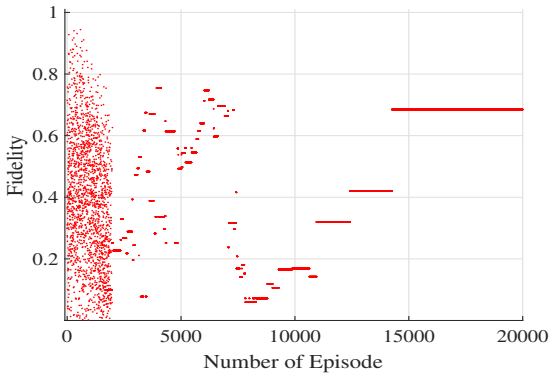

Fig. 2. Final fidelity of each episode for DQL_1 at N=20

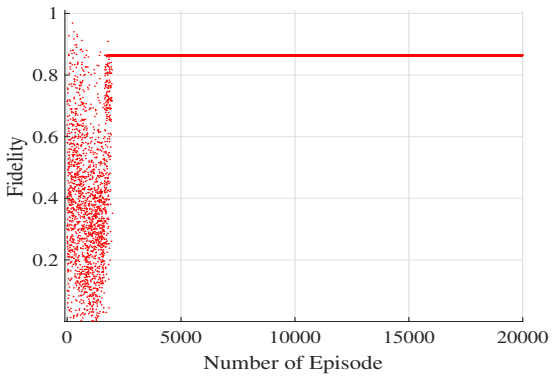

Fig. 3. Final fidelity of each episode for DQL_2 at N=20

Fig. 4 shows average fidelity of MDQL for $N = 20$ control pulses. Average fidelity has been calculated for 2000 episodes. MDQL shows improved performance of converging to the best policy that achieves the highest fidelity. In summary, MDQL can assist the RL agent to find better results and to converge to the discovered global optimal results. The MDQL can also prevent the algorithm sticking to the local optimal solution.

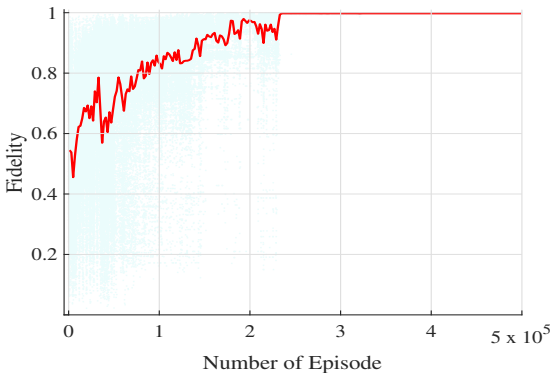

Fig. 4. Average fidelity of MDQL at N=20

In short, MDQL algorithm has succeeded for solving state preparation problem of a two-qubit closed quantum system with limited control resources and shows improved performance comparing with classical DQL algorithms. This method has the potential ability to be reliable for solving more complex problems.

V. Conclusion

In this work, we considered the control problem of a closed two-qubit quantum system with limited control resources. The aim is to drive the system into a desired target state with high fidelity. We proposed the MDQL algorithm which is a novel modified version of the DQL. A modified action selection procedure and an improved replay memory were proposed. Numerical results show that the MDQL has a high effectiveness for solving the quantum control problem with a high final fidelity, the increasing robustness and the ability to converge to the best policy. We hope this novel approach can be used to solve more complex quantum control problems.

References

[1] C. Sayrin, I. Dotsenko, X. Zhou, B. Peaudecerf, T. Rybarczyk, S. Gleyzes, P. Rouchon, M. Mirrahimi, H. Amini, M. Brune, and J. M. Raimond, "Real-time quantum feedback prepares and stabilizes photon number states," *Nature,* vol. 477, no. 7362, pp. 73-77, 2011.

[2] M. A. Nielsen, and I. L. Chuang, "Quantum computation and quantum information," *Mathematical Structures in Computer Science,* vol. 17, no. 6, p. 1115, 2007.

[3] M. Nagy, and S. G. Akl, "Quantum computation and quantum information," *The International Journal of Parallel, Emergent and Distributed Systems,* vol. 21, no. 1, pp. 1-59, 2006.

[4] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature,* vol. 574, no. 7779, pp. 505–510, 2019.

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

[6] P. Lam, and T. Ralph, "Quantum cryptography: Continuous improvement," *Nature Photonics,* vol. 7, no. 5, pp. 350-352, 2013.

[7] 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.

[8] 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.

[9] C. B. Zhang, D. Dong, and Z. H. Chen, "Control of non-controllable quantum systems: a quantum control algorithm based on Grover iteration," *Journal of Optics B: Quantum and Semiclassical Optics,* vol. 7, no. 10, pp. S313-S317, 2015.

[10] Q. Gao, D. Dong, and I. R. Petersen, "Fault tolerant quantum filtering and fault detection for quantum systems," *Automatica,* vol. 71, pp. 125-134, 2016.

[11] S. Kuang, D. Dong, and I. R. Petersen, "Lyapunov control of quantum systems based on energy-level connectivity graphs," *IEEE Transactions on Control Systems Technology,* vol. 27, no. 6, pp. 2315-2329, 2018.

[12] A. Garon, S. Glaser, and D. Sugny, "Time-optimal control of SU(2) quantum operations," *Physical Review A,* vol. 88, no. 4, p. 043422, 2013.

[13] N. Khaneja, T. Reiss, C Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, "Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms," *Journal of Magnetic Resonance,* vol. 172, no. 2, pp. 296-305, 2005.

[14] 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.

[15] 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.

[16] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature,* vol. 549, no. 7671, pp. 195–202, 2017.

[17] J.A. Li, D. Dong, Z. Wei, Y. Liu, Y. Pan, F. Nori, and X. Zhang, "Quantum reinforcement learning during human decision-making," *Nature human behaviour,* vol. 4, no. 3, pp. 294-307, 2020.

[18] 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.

[19] 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.

[20] 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.

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

[22] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE transactions on neural networks and learning systems,* vol. 29, no. 6, pp. 2216-2226, 2018.

[23] T. R. N, and R. Gupta, "A Survey on machine learning approaches and its techniques," *IEEE International Students Conference on Electrical,Electronics and Computer Science,* pp. 1-6, 2020.

[24] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, "Reinforcement learning in different phases of quantum control", *Physical Review X,* vol. 8, no. 3, p. 031086, 2018.

[25] A. G. R. Day, M. Bukov, P. Weinberg, P. Mehta, and D. Sels, "Glassy phase of optimal quantum control," *Physical Review Letters,* vol. 122, no. 2, 2019.

[26] C. Chen, D. Dong, H.X. Li, J. Chu, and T.J. Tarn, "Fidelity-based probabilistic Q-learning for control of quantum systems," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 25, no. 5 pp.920-933, 2014.

[27] 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.

[28] P. Sgroi, G. M. Palma, and M. Paternostro, "Reinforcement learning approach to non-equilibrium quantum thermodynamics," *Physical Review Letters,* vol. 126, no. 2, p.020601, 2021.

[29] 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.

[30] 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.

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

[32] H. Ma, D.Dong, S.X. Ding, and C. Chen, "Curriculum-based deep reinforcement learning for quantum control," *arXiv preprint arXiv:2012.15427*, 2020.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.