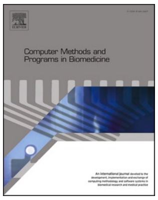
Computer Methods and Programs in Biomedicine 243 (2024) 107884

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb



# Deep reinforcement learning-based control of chemo-drug dose in cancer treatment

Hoda Mashayekhi [a], Mostafa Nazari [b,*], Fatemeh Jafarinejad [a], Nader Meskin [c]

[a] *Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran*
[b] *Faculty of Mechanical Engineering, Shahrood University of Technology, Shahrood, Iran*
[c] *Faculty of Electrical Engineering, Qatar University, Doha, Qatar*

ARTICLE INFO

*Keywords:*
Cancer chemotherapy
Active drug dosing
Deep reinforcement learning
Twin-delayed deep deterministic policy gradient
Actor-critic
Optimal control

ABSTRACT

*Background and objective:* Advancement in the treatment of cancer, as a leading cause of death worldwide, has promoted several research activities in various related fields. The development of effective treatment regimens with optimal drug dose administration using a mathematical modeling framework has received extensive research attention during the last decades. However, most of the control techniques presented for cancer chemotherapy are mainly model-based approaches. The available model-free techniques based on Reinforcement Learning (RL), commonly discretize the problem states and variables, which other than demanding expert supervision, cannot model the real-world conditions accurately. The more recent Deep Reinforcement Learning (DRL) methods, which enable modeling the problem in its original continuous space, are rarely applied in cancer chemotherapy.

*Methods:* In this paper, we propose an effective and robust DRL-based, model-free method for the closed-loop control of cancer chemotherapy drug dosing. A nonlinear pharmacological cancer model is used for simulating the patient and capturing the cancer dynamics. In contrast to previous work, the state variables and control action are modeled in their original infinite spaces to avoid expert-guided discretization and provide a more realistic solution. The DRL network is trained to automatically adjust the drug dose based on the monitored states of the patient. The proposed method provides an adaptive control technique to respond to the special conditions and diagnosis measurements of different categories of patients.

*Results and conclusions:* The performance of the proposed DRL-based controller is evaluated by numerical analysis of different diverse simulated patients. Comparison to the state-of-the-art RL-based method, which uses discretized state and action spaces, shows the superiority of the approach in the process and duration of cancer chemotherapy treatment. In the majority of the studied cases, the proposed model decreases the medication period and the total amount of administrated drug, while increasing the rate of reduction in tumor cells.

## 1. Introduction

The repeated and uncontrolled division of abnormal cells is named cancer and the corresponding tissues are called tumors. American Cancer Society has reported 606,520 deaths from a total of 1806,590 diagnosed new cancer cases in 2020 [1]. Based on this report, the survival rate has been improved significantly due to earlier diagnosis and better treatment methods.

There exist several solutions for cancer treatment such as surgery, chemotherapy, radiotherapy, immunotherapy, hormone therapy [2], anti-angiogenic therapy [3], targeted therapy [4], monoclonal antibody therapy [5], etc. Treatment type selection depends on many factors such as the patient's age and health, the stage of cancer, and its type [6]. In many cases, it is shown that mixed treatments such as chemo-immunotherapy [7], mixed monoclonal antibody therapy and chemotherapy [8], and chemo-radiation therapy [9,10] are more effective. Chemotherapy kills both healthy and cancer cells, and affects the whole body of the patients [11]. Physicians employ chemotherapy alone or along with other strategies for cancer treatment. In some cases, the cancer cells spread to another organ through the bloodstream or lymphatic system in the metastasis process [10]. The second organ may be the liver, brain, or lung and in this process chemotherapy is essential.

* Corresponding author.
*E-mail addresses:* hmashayekhi@shahroodut.ac.ir (H. Mashayekhi), nazari_mostafa@shahroodut.ac.ir (M. Nazari), jafarinejad@shahroodut.ac.ir (F. Jafarinejad), nader.meskin@qu.edu.qa (N. Meskin).

https://doi.org/10.1016/j.cmpb.2023.107884
Received 13 December 2022; Received in revised form 15 October 2023; Accepted 23 October 2023
Available online 24 October 2023

Due to the side effects of chemo-drugs, it is essential to have an optimal strategy for their intervention.

The treatment schedule and the maximum drug dose depend on many factors such as the age and weight of the patient, the ability of the immune system, and the contemporary disease of the patient. Oncologists use standard protocols for the treatment procedure and the maximum drug dose. However, these standard protocols have many limitations [12] and hence, deriving an optimal methodology for the treatment strategy is motivated by scientists and researchers. Dynamic Treatment Planning Regimens (DTR) is the strategy of deciding the personalized therapy, including dose or treatment schedule [13]. It is an adaptive treatment strategy that considers the patient's clinical information and diagnosis measurement to generate a treatment procedure. Thus, instead of providing the same kind of treatments for every patient, the customized decisions in DTR consider improving the long-term health outcomes of different patients.

Any strategy that improves the benefits of the chemotherapy treatment and reduces its side effects is greatly desired. The effectiveness of a chemotherapy plan should be evaluated and its feasibility must be checked [14]. Experimental studies in this area have a high cost and depend on numerous time-consuming tests and trials. Hence, modeling and control of the dynamics of the tumor-immune system have recently become more compelling, and more effective strategies have been developed for cancer chemotherapy by using dynamical mathematical models in cancer pharmacology [15]. Several research works present mathematical models for the interaction between cancer and immune cells by using *in silico* experimental trials [16]. Then, novel control methods are developed based on these models for deriving effective treatment strategies [3,11,17,18].

Many mathematical models are available for cancer dynamics while a convenient model for the development of cancer therapy should consider cancer cells, immune cells, and the effect of external drugs on cells. Moreover, any developed therapy not only should reduce the tumor cells but also minimize the side effects of the drugs. Among such side effects is the weakening of the immune system, which leads to life-threatening infections. The interaction among cells in cancer dynamics is complex, nonlinear, and uncertain and consequently, an open-loop conventional cancer treatment method cannot achieve an acceptable performance in the presence of nonlinearity and uncertainty of cancer dynamics. However, a closed-loop system is generally more robust concerning parameter variation and model uncertainty and hence can lead to better therapy performance. Based on the clinical response of the patient during treatment, a closed-loop control approach can change the required drug administration to account for the discrepancy between the system response and the desired response [18].

Several optimization methods are proposed for cancer chemotherapy [19]. Chen et al. [12] apply the MPC method for optimizing the chemo-drug dose for a given sampling period. The model is adjusted by measuring the state transition. The chemo-immunotherapy treatment is used in [19] where a multi-objective optimization method for optimizing chemotherapy in dealing with immunotherapy is considered. In [20], the model predictive control (MPC) method with parameter estimation is considered and Engelhart et al. [21] examine different objective functions along with four ordinary differential equation (ODE) cancer models for investigating the optimal cancer chemotherapy. Nonlinearity in cancer dynamics is one of the challenging problems in cancer control. In [22], the state-dependent Riccati equation (SDRE) method is used which has flexibility in design and is robust in dealing with parameter changes. A patient-specific controller is designed in [7] using the SDRE-based model reference adaptive control (MRAC-SDRE) method. Specific conditions of patients are considered by choosing different weighting matrices in the SDRE method [22,23].

Other optimization approaches such as genetic and other evolutionary algorithms, and computer modeling are also used to automate chemotherapy [24]. In these methods, global optimal solutions are found by following the principles of natural selection. However, the main drawbacks of these methods involve difficulty in selecting the initial population, setting parameter values of the initial population, and the computational cost [14].

### 1.1. Related works

Reinforcement learning is one of the most practiced techniques in the field of machine learning [25]. Inspired by psychology, RL enables an agent to learn from the experiences obtained through interacting with its environment [26]. To find an optimal strategy, the agent explores the space of possible strategies and receives feedback while making different choices. An ideally optimal policy (strategy or controller) may be derived by trying to maximize the cumulative performance. The proposed RL-based controller is model-free and the learned strategy is exploited to adaptively control the drug dose without any mathematical model of the patient. Instead of detailing the solution, the designer of a control task should provide appropriate rewards, which evaluate the propriety of the actions chosen by the agent. RL can address analytically intractable problems, using approximations and data-driven techniques.

RL-based methods are used for the closed-loop control of drug dosing in chemotherapy treatments [27,28], radiotherapy [29], insulin dosage [30,31], and medical decision support systems [32,33]. The treatment of anemia using RL-based control is shown in [34] where the dosage of erythropoietin is considered as the action. Optimization of the anesthetic drug infusion for surgical patients by using RL-based control is presented in [35]. Significant results in the control of Propofol infusion are derived by using the RL-based controller [18]. In [3], RL is proposed for the control of tumor growth under anti-angiogenic therapy.

To reduce the computational cost of the RL techniques, the state-action space has to be discretized. For example, in [18] an RL chemotherapy treatment control scheme is developed based on four states and one finite and discrete action. Nevertheless, to obtain a higher efficiency in chemotherapy control, it is necessary to work in a continuous state-action space. This enables the model to better emulate the real-world conditions of the problem while removing the necessity of expert supervision to determine appropriate discretization rules. For example, in [36], an integral RL is used to deal with continuous control of propofol drug dosage. The Actor-Critic (AC) model [37] is an online policy method, which uses two different networks, called actor and critic. In AC methods, the policy, known as the actor, can be updated through the deterministic policy gradient algorithm, which is then used in the critic to update the value function according to the direction suggested by the actor.

DRL methods are powerful model-free approaches to deal with complex systems, in the original continuous action and state spaces. The success of the Deep Q-Networks approach [38] generated extensive research and implementation of RL techniques to address high-dimensional and continuous problems within the dynamic systems control area [39]. The popularity and significant achievements of deep neural networks (DNN) [40] have motivated the generation of different variants and improvements of these networks. Mnih et al. [38] introduced the deep Q-Network (DQN) which approximates the value function for actions with a convolutional neural network (CNN), instead of a table to expand the size of the problems that can be solved with RL. However, it can only deal with systems with continuous state space but with finite and discrete action spaces. In [41], the double DQN technique was proposed to use both a value function and a new function called the advantage function, which represents the advantage of choosing an action in a state. In fact, in a doubled DQN, there exist two deep networks for the value and advantage which are combined to form the value function. The deterministic policy gradient methods, e.g. value-based method of DQN, can be used much more efficiently than the usual stochastic policy gradient models such as AC. Deterministic policy gradient (DPG) [42] uses another idea to learn a deterministic policy where it computes the gradient of expected return and updates its parameters through the gradient ascent. Lillicrap et al. [43] extended DRL formulations to continuous state spaces, proposing the deep deterministic policy gradient algorithm (DDPG) as a model-free off-policy algorithm combining the advantages of AC [37], DPG [42], and DQN [44] where batch normalization [45] and repetition of experiences [38] are incorporated. Recently, more approaches to address control problems dealing with continuous spaces have been proposed [46–49]. In [50], the DRL approach has been used for cancer radiotherapy.

In this paper, the cancer chemotherapy control problem is addressed as an optimization problem and is solved using a DRL-based method. A nonlinear pharmacological cancer model is used for simulating the patient and capturing the cancer dynamics. It should be noted that the purpose of this model is to provide just a simulation environment for training the reinforcement agent. However, the resulting learned controller will be a model-free one exploited to adaptively control the drug dose without any mathematical model of the patient. In contrast to many available approaches, which discretize the problem variables, we model the system states and the drug dosage in their original continuous spaces. A drug dosing controller is proposed which trains a DRL network to automatically administrate the drug dosage based on the monitored states of the patient. Avoiding the discretization of variables enables a more accurate modeling of the real-world conditions of the patient while decreasing the need for expert supervision. To the best of our knowledge, few works used DRL techniques to administration of chemo-drug doses for cancer therapy. For example, in [51], the DRL approach has been studied for cancer chemotherapy considering a multi-criteria decision-making strategy. In this study, the authors consider the tumor cell population and effector cell population as inputs to propose a personalized treatment strategy. In our proposed strategy, different reward functions based on the special conditions of the patients are considered. Moreover, three different patients with diverse conditions are considered to validate and evaluate the performance of our proposed method. Comparison with previous work shows the superiority of our approach in prescribing less drug dosage while imposing shorter treatment duration. The contributions of this work are summarized as follows:

- Proposing a deep RL-based controller for cancer chemotherapy that handles states and action in infinite space.
- After the initial learning phase, the proposed approach does not need to have a mathematical model of the system, and operates based on the patient's conditions.
- Providing an adaptive control technique to respond to the special conditions and diagnosis measurements of different categories of patients.
- Evaluating and comparing the efficiency and robustness of the proposed method using patients with diverse conditions.

The contents of the paper are as follows. In Section 2, the nonlinear pharmacological cancer model is described. Section 3 presents our proposed DRL-based optimal controller. The results and discussions are elaborated in Section 4. Finally, Section 6 presents the conclusion and future work.

## 2. Methods

### 2.1. Cancer mathematical model

There are many mathematical models to capture the tumor-immune interaction dynamics [52–54]. In this paper, the four-state model presented in [53] is used to show the performance of the proposed DRL-based controller. In the chosen model, the tumor-immune interaction has been studied by considering both the innate immune system and the adaptive immune system. Moreover, this model is experimentally validated and has been used in other papers. Hence, the comparison of the proposed strategy for optimal chemotherapy with other papers is possible.

The four states of the system are normal cells *N(t)* , tumor cells *T(t)*, immune cells *I(t)*, and the concentration of the chemo-drug *M*(*t*). By defining the state variables as $x_1(t) = N(t)$, $x_2(t) = T(t)$, $x_3(t) = I(t)$, and $x_4(t) = M(t)$, the state-space model of tumor-immune interaction dynamics is given by:

$$\dot{x}_1(t) = r_2 x_1(t)\ (1 - b_2 x_1(t)) - c_4 x_1(t) x_2(t) - a_3 x_1(t)\left(1 - e^{x_4(t)}\right) \tag{1}$$

$$\dot{x}_2(t) = r_1 x_2(t)\ (1 - b_1 x_2(t)) - c_2 x_3(t) x_2(t) - c_3 x_1(t) x_2(t) - a_2 x_2(t)\left(1 - e^{x_4(t)}\right) \tag{2}$$

$$\dot{x}_3(t) = s + \frac{\rho x_3(t) x_2(t)}{\beta + x_2(t)} - c_1 x_3(t) x_2(t) - d_1 x_3(t) - a_1 x_3(t)\left(1 - e^{x_4(t)}\right) \tag{3}$$

$$\dot{x}_4(t) = -d_2 x_4(t) + u(t) \tag{4}$$

where the growth of the normal cells and cancer cells are considered as a logistic term with rates $r_1$ and $r_2$, respectively, the coefficients $b_1$ and $b_2$ show the reciprocal of the carrying capacity for normal cells and cancer cells, and the interaction term between cells is modeled as a product form with different competition rates. The type of the model shows that when the immune cells are large in number, the existence of tumor cells is low and vice versa [52]. Immune cells proliferate to create new cells and die after their lifetime with rate $d_1$, and the influx rate of immune cells is regarded as a constant *s*. The immune cells are also proliferating due to the existence of tumor cells, and this phenomenon has a saturation limit and is incorporated in the model with the term $\frac{\rho x_3(t) x_2(t)}{\beta + x_2(t)}$ in (3), where β and ρ are positive constants. The term *u*(*t*) represents the drug infusion rate. The chemo-drug not only destroys the cancer cells but also annihilates immune cells and normal cells. The target of the controller is to choose the optimal dosage of the chemo-drug to reach two targets: minimizing the chemo-drug dose and eradicating cancer cells.

### 2.2. DRL-based optimal controller design

Reinforcement learning (RL) is a paradigm for learning optimal behavior in unknown environments. The RL algorithm aims to learn the behavior of a system or its optimal configuration according to the responses of the interactions with the environment. It remembers the reactions of the environment (in the form of reward or punishment) to the behaviors of the agent, to improve its future behavior. When exploring the optimal strategy with DRL, new experiences are learned through trial and error. While interacting with the environment, a DRL agent attempts to learn actions that maximize the cumulative reward obtained. One of the biggest challenges of RL algorithms is dealing with spaces of continuous state and action. Although a common approach is discretizing such spaces, it may end with a dimensionality problem. In addition, discretization of the space can neglect valuable information about the domain geometry. DRL can deal with continuous spaces by changing the representation of the action-value function. In what follows, we first briefly describe RL and DRL in two subsections, followed by the description of the TD3 method and finally the proposed controller.

#### 2.2.1. Reinforcement learning

Generally, RL problems are modeled and solved iteratively using Markov decision processes (MDPs) theory through Markov Chain Monte Carlo (MCMC) and dynamic programming (DP) [26]. The four sequences of finite set of states $\mathscr{S}$, a finite set of actions $\mathscr{A}$, the probability of transition from state *s* to state $s'$ under action *a* at time *t*, i.e. $P_a(s,s') = \Pr(s_{t+1} = s' | s_t = s,\ a_t = a)$, and the immediate reward after transition from *s* to $s'$ with action, $R_a(s,s')$, are used in the finite MDP framework to capture the system dynamics.

The scenario of the behavior of an RL agent in the environment is as follows: At each time *t*, the agent receives the current states$_t$ and reward$r_t$. It then chooses an action $a_t$ from the set of available actions and performs it. Thereafter, the environment moves to a new state $s_{t+1}$ and the reward $r_{t+1}$, associated with the transition $(s_t, a_t, s_{t+1})$, is given to the agent. The appropriate reward $r_{t+1} \in \mathbb{R}$ shows the desirability of the selected action $a_t$ [16]. The action selection is modeled as a map called policy. The goal of an RL agent is to learn a policy, $\pi(s, a) = Pr(a_t = a | s_t = s)$, which maximizes the expected cumulative reward. Cumulative reward in time $t$ is computed as:

$$R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r(s_i, \ a_i) \tag{5}$$

where $r(s_i, a_i)$ is the reward of choosing $a_i$ in state $s_i$, and $\gamma$ is the discount factor in the range [0,1] which is used to prefer the reward of the current action from the future rewards. Hence, the goal of RL is to maximize the expected return, $J = \mathbb{E}[R_0]$ which can be achieved based on the stochastic techniques or the deterministic ones.

There are different criteria for optimality of agent behaviors. In the Q-Learning (QL) algorithm, $Q^{\pi}(s,a)$ is a quality function, which represents the quality of choosing an action in a specific state. The optimal $Q$-function $Q^*(s, a)$ is defined as the maximum return that can be obtained starting from state $s$, taking an action $a$ and following the optimal policy afterwards. $Q^*(s, a)$ conforms to the Bellman optimality equation given as

$$Q^*(s,a) = \mathbb{E}\left[R_a(s,s') + \gamma \max_{a'} Q^*(s',a')\right] \tag{6}$$

where the maximum return from state $s$ and action $a$, is the sum of the immediate reward and the maximum reward from the next state $s'$ by following the optimal policy.

The $Q$-function is computed iteratively, and in each iteration, $Q(s, a)$ is recalculated by averaging the reward of action $a$ in the state $s$:

$$Q(s,a) = \ Q(s,a) + \alpha\left[R_a(s,s') + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{7}$$

where $\alpha$ is the learning rate. The $\varepsilon$-greedy technique is a common method to fill the $Q$-table of a QL agent, which balances between exploration (choosing new actions in the next state) and exploitation (choosing the best next action). This technique chooses the best action (action corresponding to $\operatorname{argmax}_a Q(s,a)$) with the probability $1 - \varepsilon$, and randomly chooses an action with the probability $\varepsilon$.

### 2.2.2. *Deep reinforcement learning*

In problems with a large number of states, it is very time-consuming to fill the $Q$-table of Q-Learning. Deep Q-network (DQN) [44] represents the optimal action-value function as a deep neural network ($Q$-network) with parameters (or weights) $\theta$, instead of a table, to expand the size of the problems that can be solved with RL. Using $Q$-network allows for dealing with states and actions with continuous values, without the need to discretize the space. The $Q$-network acts as a function approximator to estimate the $Q$-values, i.e. $Q(s, a; \theta) \approx Q^*(s, a)$. It can be trained by minimizing a sequence of loss functions $L_i(\theta_i)$ that are changed at each step $i$:

$$L_i(\theta_i) = \ \mathbb{E}_{s,a\sim\rho()}\left[(y_i - Q(s,a;\theta_i))^2\right] \tag{8}$$

where $y_i = \mathbb{E}\left[R_a(s,s') + \gamma \max_{a'} Q(s',a';\theta_{i-1})\right]$ is the TD (temporal difference) target for iteration $i$, $y_i - Q(s, a; \theta_i)$ is the TD error, and $\rho(s, a)$ is the behavior distribution on states $s$ and action $a$, which is often selected by the $\varepsilon$-greedy strategy. It is worth noting that in contrast to targets used for supervised learning, which are fixed, here the targets depend on the network weights. Rather than computing the full expectation, the loss function is often optimized using the stochastic gradient descent.

There are other differences between DQN and QL. Considering the update equation of the $Q$ -function in (7), during the learning it is aimed that $Q(s, a)$ approaches to the $R + \gamma \max_{a'} Q(s', a')$. Utilizing an identical network, $Q$, in both $Q(s, a)$ and $Q(s', a')$, the weight update in training the deep network will change both values $Q(s, a)$ and $R + \gamma \max_{a'} Q(s', a')$, simultaneously. Therefore, DQN uses a different target network for the next action, $Q(s', a')$. Furthermore, it is an offline policy learning about the greedy strategy while following a behavior distribution that ensures adequate exploration of the state space. The Experience Replay technique is also introduced in DQN, in which the transitions are added to a circular replay buffer. When training, instead of using just the most recent transition to compute the loss and its gradient, a mini-batch of transitions sampled from the replay buffer is used. This ensures better data efficiency by reusing each transition, and better stability using uncorrelated transitions in a batch.

QL algorithms suffer from the problem of over-estimation that is propagated during training iterations and affects the trained policy. Doubling is another idea that decouples the action selection and the $Q$-value update procedures into two separate networks [55]:

$$Q_A(s,a) = \ Q_A(s,a) + \alpha\left[R + \gamma \max_{a'} Q_B\left(s', \operatorname{argmax}_a \ Q_A(s',a)\right) - Q_A(s,a)\right] \tag{9}$$

$$Q_B(s,a) = \ Q_B(s,a) + \alpha\left[R + \gamma \max_{a'} Q_A\left(s', \operatorname{argmax}_a \ Q_B(s',a)\right) - Q_B(s,a)\right] \tag{10}$$

Double DQN [41] uses the idea of doubling the $Q$-network. However, as the DQN already has two different networks, Double DQN uses the additional target network to double the DQN. Moreover, Double DQN proposes the technique of dueling DQN. This idea decomposes the $Q$-function into the value function, $V(s)$, and the advantage function $A(s, a)$, which represents the advantage of choosing the action $a$ in the state $s$. In fact, in a dueled DQN, we have two deep networks $V(s)$ and $A(s, a)$ which are combined to form the $Q$-function:

$$Q(s,a) = V(s) + A(s,a) \tag{11}$$

The Actor critic (AC) model [37] is an online policy method that uses an idea that is very similar to the dueling technique in [41]. It uses two different networks, called the actor and critic. However, unlike the dueling technique, these networks are not combined at the end. In the AC model, the policy, known as the actor, can be updated through the deterministic policy gradient algorithm. It is then used in the critic to update the Q function (or the value function) according to the direction suggested by the actor. The actor in the AC model uses a stochastic policy to assign probabilities to each action. This procedure only works on-policy. On the other hand, successful deterministic policy gradient methods, e.g. value-based methods in DQN, can be estimated much more efficiently than the usual stochastic policy gradient models such as AC. Deterministic policy gradient (DPG) [42] uses another idea to learn a deterministic policy. It computes the gradient of expected return, $J$, and updates its parameters through the gradient ascent.

Deep deterministic policy gradient (DDPG) [38,43] is a model-free off-policy algorithm combining the advantages of AC, DPG, and DQN. It outperforms AC in high-dimensional state spaces and uses fewer learning samples in problems with high-dimensional action spaces. It is an offline-policy method that can be utilize from an experience replay memory. DDPG uses the deterministic policy gradient just as in DPG. However, it uses the idea of DQN to apply this technique in problems with continuous high-dimensional state space. DDPG suffers from the problem of overestimation as the other Q-learning algorithms. The Twin Delayed DDPG (TD3) [56] algorithm is a generalization of DDPG that uses clipped double Q-learning to prevent this problem. Similar to double Q-learning [55], action selection and Q value estimation are assigned to two different networks. Furthermore, like DDPG, there are separate actors and critics. To overcome the issue of the slow-changing

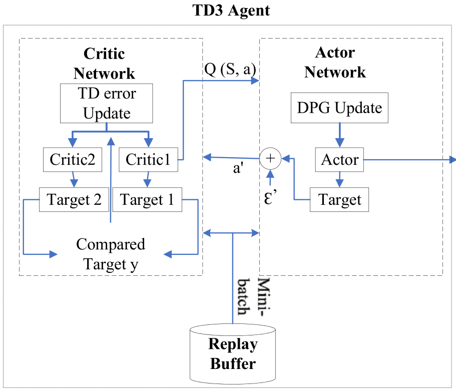


**Fig. 1.** Network structure of TD3 cancer controller.

policy, as a side effect of the similarity of doubled networks, TD3 proposes the idea of clipped double Q-learning. In this regard, it uses the minimum estimation of twined critics to favor understanding bias which is hard to propagate through training. In fact, TD3 uses six different networks: actor, target actor, two critics, and two target critics. To ensure that the error remains small, the target networks are updated slowly, every $d$ iterations. In this paper, we use the TD3 algorithm to learn an optimal strategy for controlling the chemotherapy treatment.

### 2.2.3. *Twin delayed deep deterministic policy gradient*

As stated in Section 3.2, TD3 uses a stochastic policy to achieve a good exploration and estimates a deterministic objective policy. It is built on DDPG with several modifications to increase stability and performance. TD3 is based on an actor-critic approach and uses six deep neural network models: actor, target actor, two critics, and two target critics. These networks obtain the optimal policy for choosing an action for the current state of a continuous control setting. The input to the actor-network is the current state, and the output is a single real value that represents an action chosen from the continuous action space. The output of the network that models the critic is simply the estimated Q value for the current state and the action given by the actor. TD3 agent starts with an initial arbitrary policy and this policy is updated by interacting with the system. The strategy approaches the optimal strategy as the agent receives more information in terms of states, actions, and rewards, and stores them in the experience replay buffer $\mathscr{B}$. As explained before, there are six networks: an actor model $\pi$, an actor target model $\pi'$, two critic models $Q_1$,$Q_2$, and two critic target models $Q_1^{'}$, $Q_2^{'}$, with weight parameters $\varphi$, $\varphi^{'}$, $\theta_1$, $\theta_2$, $\theta_1^{'}$, $\theta_2^{'}$, respectively. The network structure of the TD3 cancer controller illustrated in Fig. 1 shows the interaction of different components of TD3.

Starting with an empty replay buffer, the networks of the actor model and two critic models are initialized with random weight parameters. The actor target model has a similar structure as the actor model and its initial weights are the same. Similarly, the critic target models have the same structure and initial weights as the critic models. During the training, the weights of the two critic models are updated in each iteration, but the weights of the actor model and three target models are updated every $d$ iterations. As an off-policy method, in each iteration of the training process, the TD3 agent first samples a mini-batch of $N$ transitions ($s$, $a$, $s'$, $r$) from the replay buffer $\mathscr{B}$. Giving as an input the next state of each transition $s'$, to the actor target model, the next action will be obtained. Adding a clipped Gaussian noise ε ($\epsilon \sim clip(\mathscr{N}(0,\sigma, -c, c), \ c > 0)$, the action is obtained as follows:

$$\widetilde{a} = \pi^{'}(s^{'}) + \epsilon \tag{12}$$

The clip function limits εto belong to the interval $[-c, c]$. Then, the couple $(s^{'}, \ \widetilde{a})$ is given to each of the two critic targets. The minimum of outputs of these critic networks is used as an approximation of the best quality function of the next action. Based on this, a target value $y$, is computed as follows:

$$y \leftarrow r + \gamma \min_{i=1,2} Q_i^{'}(s^{'}, \widetilde{a}) \tag{13}$$

During the learning phase, it is aimed that $Q_i(s,a)$ approaches to $y$. Hence, the critic loss of each of the two critics is computed as: MSE($Q_1$($s$, $a$), $y$) + MSE($Q_2$($s$,$a$), $y$), where MSE is the Mean Square Error function. This loss is used to update the parameters of critic models using the backpropagation strategy in gradient descent. For every $d$ iterations, the parameters of the actor model are updated using the gradient ascent on the output of the first critic model, as follows:

$$\nabla_{\varphi} J(\varphi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_{\varphi}(s)} \nabla_{\varphi} \pi_{\varphi}(s) \tag{14}$$

For every $d$ iterations, the critic target networks parameters are updated by taking the weighted average of their weights and

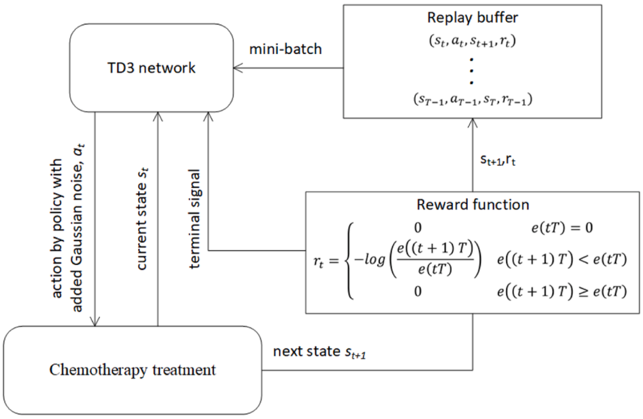

Replay buffer
$(s_t, a_t, s_{t+1}, r_t)$
$(s_{T-1}, a_{T-1}, s_T, r_{T-1})$
mini-batch
TD3 network
$s_{t+1}, r_t$
Reward function
$r_t = \begin{cases} 0 & e(tT) = 0 \\ -log\left(\frac{e((t+1)T)}{e(tT)}\right) & e((t+1)T) < e(tT) \\ 0 & e((t+1)T) \geq e(tT) \end{cases}$
action by policy with added Gaussian noise, $a_t$
current state $s_t$
terminal signal
Chemotherapy treatment
next state $s_{t+1}$

**Fig. 2.** The process of chemotherapy treatment control.

**Algorithm 1**
The chemotherapy treatment policy modeling.

```
Input: initial_state [$x_1$,$x_2$,$x_3$,$x_4$], terminal-condition
  Output: Trained optimal policy
  1. Initialize hyperparameters: batch-size, N, d, τ, ε, γ
  2. Initialize TD3:
  3. Initialize the experience replay buffer $\mathscr{B}$
  4. Initialize actor model $\pi_\varphi$ with random parameters φ, and initialize actor target
  network$\pi^{'}_{\varphi^{'}}$ with parameters φ′ ← φ.
  5. Initialize critic models $Q_{\theta1}$,$Q_{\theta2}$ with random parameters $\theta_1$,$\theta_2$, and initialize critic
  target network $Q^{'}_{\theta^{'}_1}$ , $Q^{'}_{\theta^{'}_2}$ with $\theta^{'}_1 \leftarrow \theta_1$, $\theta^{'}_2 \leftarrow \theta_2$.
  6. While training-episodes are not finished, do:
  7. Reset state to the initial state of ($x_1$,$x_2$,$x_3$,$x_4$)
  8. While goal-state or max-steps are not reached, do:
  9. Select action for the current state according to epsilon-policy and add noise
  10. Compute the next-state according to Eqs. (1)-(4)
  11. Compute reward function according to Eq. (15)
  12. Add transition (state, next-state, action, reward) into $\mathscr{B}$
  13. Move to the next state
  14. Sample a mini-batch of N transitions (s, s′, a, r) from the replay buffer $\mathscr{B}$.
  15. Generate the next action $\widetilde{a}$ based on s′ using actor target DNN, add clipped
  Gaussian noise (Eq. (12)).
  16. Give as input the tuple $(s^{'}, \ \widetilde{a})$ to each of the two critic targets, to gain their
  minimum output as an approximation of the best quality function of the next action,
  y (Eq. (13)).
  17. Give as input the tuple(s, a) to each of the two critics, and compute the critic loss
  as$c_{loss}$ = MSE($Q_1$(s,a), y) + MSE($Q_2$(s,a), y).
  18. Back propagate the $c_{loss}$ and update the parameters of critic models using
  gradient descent.
  19. In every d iterations, update the actor model by gradient ascent on the output of
  the first critic model (Eq. (14)).
  20. In every d iterations, update parameters of target networks by averaging their
  weights and corresponding network weights: $\theta^{'}_i \leftarrow \tau\theta_i + (1-\tau)\theta^{'}_i$, φ′ ← τφ + (1 − τ)φ:
  21. Apply epsilon decaying after warm-up episodes
```

corresponding critic weights as $\theta_i^{'} \leftarrow \tau\theta_i + (1-\tau)\theta_i^{'}$. Moreover, the parameters of the actor target network are updated by averaging its weights and corresponding actor weights as $\varphi' \leftarrow \tau\varphi + (1-\tau)\varphi$.

### 2.2.4. The proposed DRL-based controller

This section presents the proposed model-free TD3 approach to solve the problem of continuous chemotherapy treatment control. The nonlinear four-state ODE cancer model is given by (1–4). The general schematic of the proposed DRL-based controller is shown in Fig. 2 and the corresponding algorithm is given in Algorithm 1.

To formulate the chemotherapy control problem as a Markov decision problem into the DRL framework, we should define its main three elements. In this work, a continuous state space is defined consisting of the number of immune cells, normal cells, tumor cells, and the concentration of the chemo-drug$(x_1,x_2,x_3,x_4)$. In cancer chemotherapy, the best chemo-drug dose should be identified such that the initial non-zero tumor cells are pushed to the desired final state in which the tumor cells $x_2(t)$ are regulated to zero. The action is the chemo-drug dosage, i.e. $u(t)$ and each action chosen by the agent in a particular state evolves the system into a successor state. A reward value is generated corresponding to this transition and to assess the desirability of the chosen action the following reward function is used:

$$r_t = \begin{cases} -log\left(\dfrac{e((t+1)\ T)}{e(tT)}\right) & e((t+1)\ T)\langle e(tT) \\ 0 & e((t+1)\ T) \geq e(tT) \end{cases} \tag{15}$$

where $e(t), t \geq 0$ involves a particular combination of the system states as explained in Section 4 and $T$ is the sampling time. This reward function originated from the reward function proposed by Padmanabhan et al. [18], and the added Logarithm function amplifies the reward of the very low error values.

In the proposed method, the hyper-parameters as well as the parameters of the TD3 agent are first initialized. Then, in each episode of training DRL, we step through the control function until the desired goal state or a maximum number of iterations is achieved. In each step, an action is selected according to the ε-greedy strategy, i.e. with the probability of ε, an action is selected randomly; otherwise, the action suggested by the actor model is selected. Moreover, as per (12), noise is added to the action and it is clipped. The derived action is considered as the control input $u(t)$, i.e. the drug dosage. Using this and the current value of states, the calculated control input $u(t)$ is used to obtain the next states based on the dynamics 1–(4). Moreover, the reward function and termination conditions are calculated as well. At the end of each step, the data structure (state, next state, action, reward) is added to the replay buffer. After performing each step, the epsilon is decreased by the decay rate. Furthermore, the TD3 agent is trained based on the new experiments as described in Section 3.3. Finally, with the exploration of the system when $t \to \infty$, the optimal policy is derived. In most cases, convergence to the optimal policy is achieved with an acceptable tolerance.

## 3. Results

To represent the efficiency of the proposed closed-loop treatment strategy, different numerical simulations are performed. Simulations are performed on a PC with an Intel Core i7 processor running at 2.8 GHz using 12 GB of RAM. We utilized a basic implementation of TD3 on GitHub.[1] The resulting code in Python is provided in GitHub[2] as well.

The upper limit of the chemo-drug dose is limited based on the oncologists' recommendations. Generally, several factors affect the oncologists' decision for choosing the desired chemo-drug dose such as age and gender of the patient, current diseases, or other special cases such as pregnancy. For example, for a young patient, it is possible to prescribe a larger dose than an elderly patient since the body of a young patient has more ability to rebuild healthy/immune cells. While an older patient should be offered a lower dose since his/her body cannot regenerate healthy/immune cells as young ones. Therefore, the treatment strategy should be able to maintain healthy/immune cells in his/her body as much as possible. Moreover, selecting a more appropriate reward function can account for the special conditions of the patients.

**Table 1**
Parameter values used to generate simulated patients [17,53,57].

| Parameter | Parameter description | Value | Unit |
|---|---|---|---|
| $a_1$ | Fractional immune cell kill rate by chemotherapy | 0.2 | $mg^{-1}lday^{-1}$ |
| $a_2$ | Fractional tumor cell kill rate by chemotherapy | 0.3 | $mg^{-1}lday^{-1}$ |
| $a_3$ | Fractional normal cell kill rate by chemotherapy | 0.1 | $mg^{-1}lday^{-1}$ |
| $b_1$ | Reciprocal carrying capacity of tumor cells | 1 | $cell^{-1}$ |
| $b_2$ | Reciprocal carrying capacity of normal cells | 1 | $cell^{-1}$ |
| $c_1$ | Inactivation rate of immune cells by tumor cells | 1 | $cell^{-1}day^{-1}$ |
| $c_2$ | Inactivation rate of tumor cells by immune cells | 0.5 | $cell^{-1}day^{-1}$ |
| $c_3$ | Inactivation rate of tumor cells by normal cells | 1 | $cell^{-1}day^{-1}$ |
| $c_4$ | Inactivation rate of normal cells by tumor cells | 1 | $cell^{-1}day^{-1}$ |
| $d_1$ | Death rate of immune cell | 0.2 | $day^{-1}$ |
| $d_2$ | Rate of chemo-drug decay | 1 | $day^{-1}$ |
| $r_1$ | Tumor cell growth rate | 1.5 | $day^{-1}$ |
| $r_2$ | Normal cell growth rate | 1 | $day^{-1}$ |
| $s$ | Influx rate of immune cell | 0.33 | $cell^{-1}day^{-1}$ |
| β | Threshold rate of immune cell | 0.3 | cell |
| ρ | Response rate of immune cell | 0.01 | $day^{-1}$ |

In this paper, the mathematical model represented by 1–(4) is used to show the performance of the DRL-based control for chemotherapy drug dosing. In the experiments, three cases with cancer are considered, namely, (1) an adult, (2) a pregnant woman who is due in 20 days, and (3) an elderly patient with other critical illnesses, and different DRL agents are trained for each case. A studied approach to cancer chemotherapy using reinforcement learning is based on the well-known Q-learning approach [3,18,27,36]. In this case, both the state and actions should be discretized so that the Q-table can be learned appropriately. In the DRL approach, in contrast, we train a control agent, which manipulates both its input state and output action in the original continuous space. The proposed method is compared with a control agent operating based on the study of Padmanabhan et al. [18], which is based on Q-learning.

The DRL is trained between 30 and 50 episodes. Each training episode consists of a maximum number of 2000 samples with a sampling time of $T = 0.1$ day. For all cases, an ε-greedy exploration strategy was used, with a decay rate of 0.95. A repeat experience buffer with a maximum size of 1e6 was established with a random selection lot size of 1000 samples. Gaussian noise with a standard deviation of 0.2 clipped to (−0.5, 0.5) is added to the actions, and τ is set to 0.005. The parameters of the system are given in Table 1.

### 3.1. Parameter study of the DRL-based controller

We analyze some parameters of the TD3 network to select appropriate values in the experiments. By varying the values of each parameter, we observe the error value during three training processes of the network. In more efficient training, the error should be reduced more quickly and in fewer steps of drug administration. Therefore, we indicate the average number of steps during the learning phase in which the error reduces to the threshold of $1e-4$ and compare it among different

[1] https://github.com/leo27945875/TD3-Ant-v2/blob/master/TD3_Ant.ipynb
[2] https://github.com/CISLAB-SUT/DeepRLChemoDrugControl

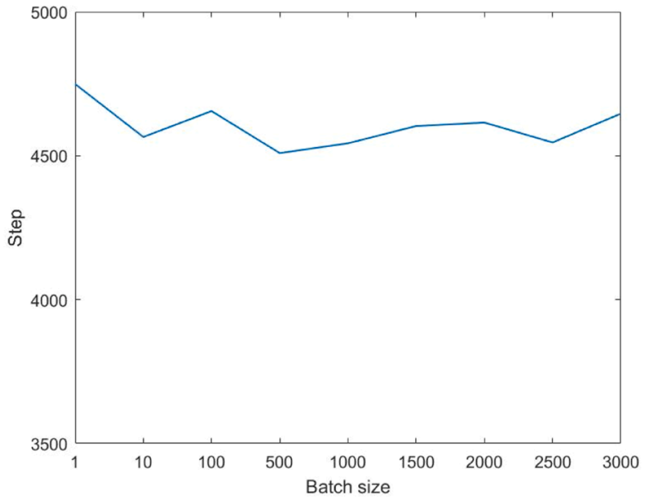


**Fig. 3.** Average steps of reaching the reduced error threshold when batch size varies.

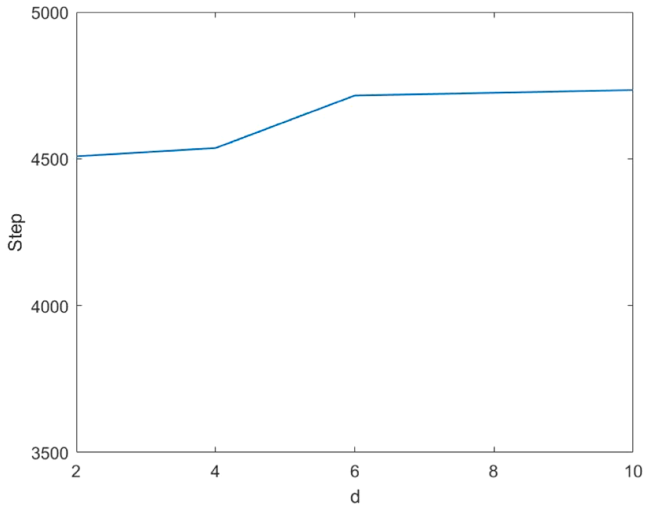


**Fig. 4.** Average steps of reaching the reduced error threshold when the policy update parameter $\boldsymbol{d}$ varies.

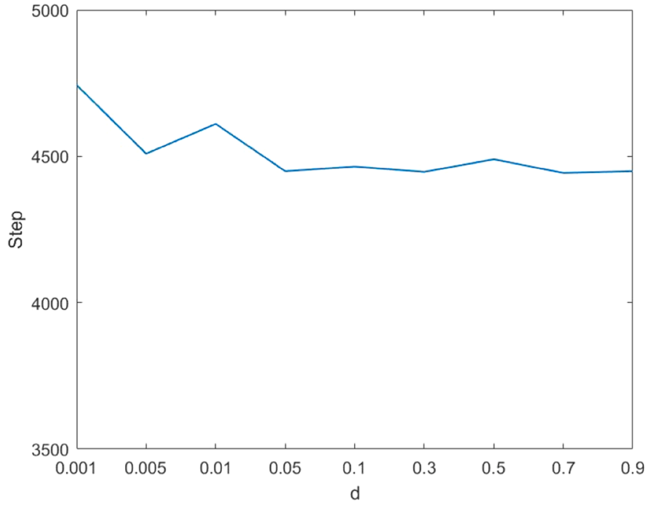


**Fig. 5.** Average steps of reaching the reduced error threshold when $\tau$ varies.

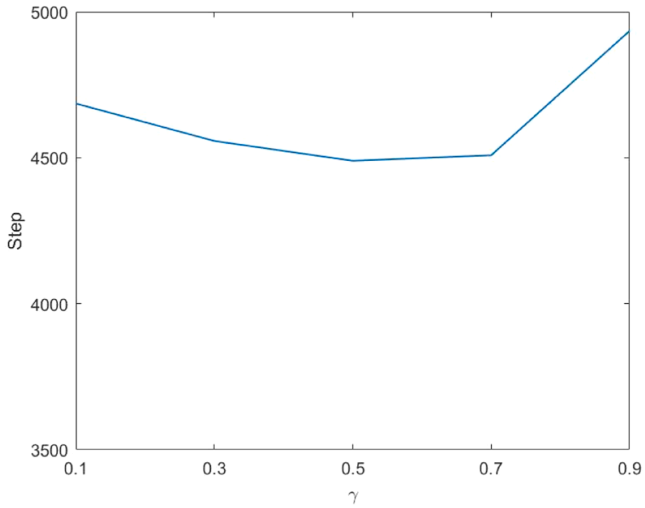


**Fig. 6.** Average steps of reaching the reduced error threshold when discount parameter $\gamma$ varies.

parameter values.

According to Section 2.2.4, the batch size determines the number of transitions sampled from the replay buffer when training the network. In Fig. 3, we change the batch size from 1 to 3000 and measure the average number of steps for the error to reach $1e-4$. As observed, very small or large values of batch size inversely affect the efficiency of the algorithm. With the batch size value of 500, the error is reduced in a smaller number of steps. This value is used in the rest of the experiments.

As described in the previous section, the weights of the actor model and three target models are updated every $d$ iterations. In Fig. 4, this parameter is varied from 2 to 10 and the effect on error reduction rate is analyzed. When the models are updated for longer periods, the step of reaching the defined error threshold increases, which means the training prefers more frequent updates to the models. We set the value of this parameter to 2 in the experiments.

For every $d$ iterations, the parameters of the critic and actor target networks are updated by taking the weighted average of their weights and the weights of their corresponding networks. This weight parameter $\tau$ is varied in Fig. 5 from very small values near zero to large values near one. Initially, the rate of error reduction is low, but it quickly gets nearly steady when $\tau$ is increased. We set this value to 0.005 in the experiments.

Another important parameter is the discount factor $\gamma$, which is used to prefer the reward of the current action over the future rewards. This value is varied from 0.1 to 0.9 in Fig. 6 and the number of steps for reaching the error threshold is reported. As observed, very low or very high discount values, which unrealistically decrease or increase the weight of early rewards, diminish the algorithm efficiency. We pick the discount factor to be 0.7 as it aids in reaching the error threshold more quickly.

### 3.2. Evaluation on different patients

In the next parts, we study the algorithm behavior for three different patients.

Case 1: A young patient

In a young patient, elimination of the tumor cells is the precedence. This is due to the ability of the body to recover the healthy and immune cells. Hence, the desired final state in this case is $x_{2d} = 0$. Therefore, the error $e(t)$can be defined as $e(t) = x_2(t) - x_{2d} = x_2(t)$. The upper limit of the chemo-drug dose is $u_{max} =$ 4.4 mg L$^{-1}$ day$^{-1}$. The response of the patient is shown in Fig. 2 when a chemotherapeutic drug is administrated based on the DRL controller and includes the plots of the number of normal cells ($x_1$), the number of tumor cells ($x_2$), the number of immune cells ($x_3$), and the concentration of chemotherapeutic drug in the

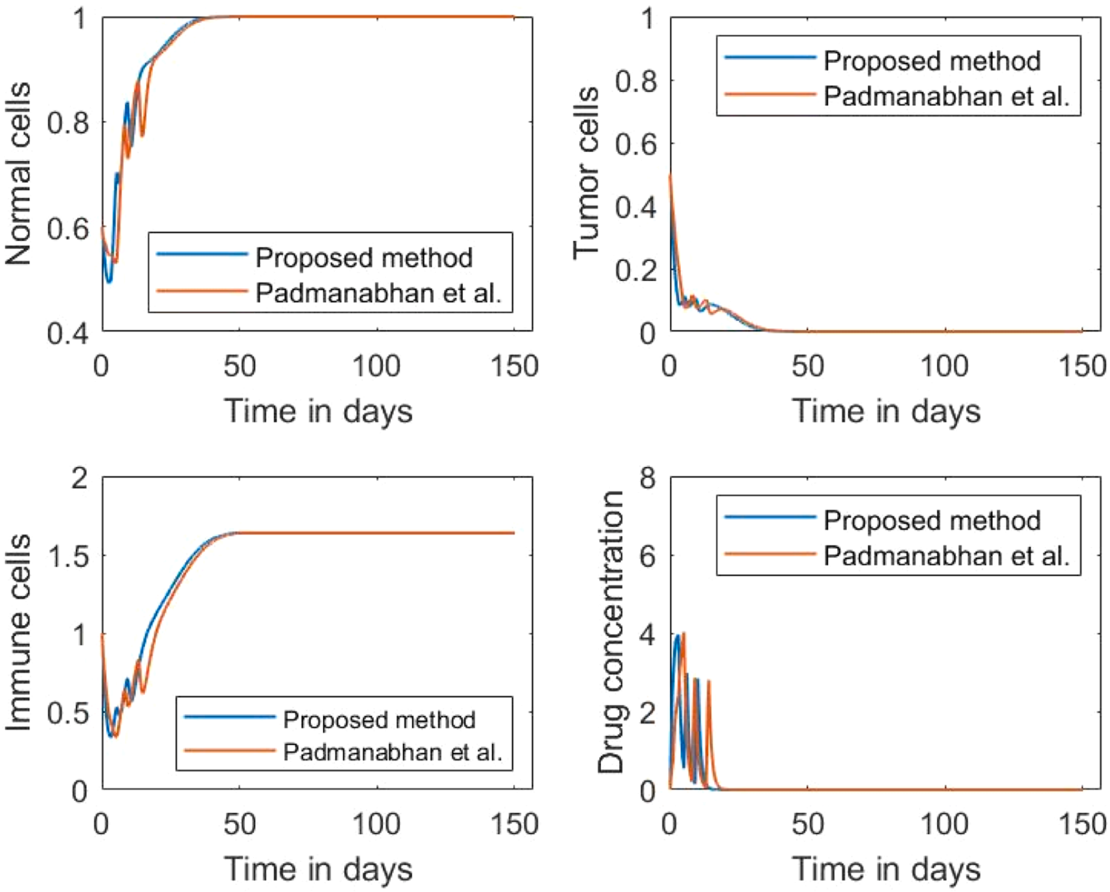


**Fig. 7.** Response of a young patient with cancer (Case 1), $\boldsymbol{u_{max}} = 4.4\ \boldsymbol{mgl^{-1}day^{-1}}$.

blood ($x_4$). As observed, the number of tumor cells decreases with time as the drug is administrated, and the number of normal cells increases. The immune cells experience an initial decrease which is due to the chemotherapy and rise afterward. The proposed DRL-based controller and the method of Padmanabhan et al. [18] expose a similar behavior in terms of different system states, but the latter has a longer medication period. This is further observed in Fig. 4, where the drug administrated for this case is plotted. The method of Padmanabhan et al. imposes a longer period of drug administration. In the proposed method, after day 10, the treatment is ceased since the trajectory of the system is in the domain of attraction of the stable tumor-free equilibrium point. In other words, from this point, the immune system can remove cancer cells without any need for external treatments.

Case 2: A young pregnant woman

In this case, the upper limit of the chemo-drug dose is maintained at the possible low level up to fetus birth. After childbirth, the upper limit of the chemo-drug dose is increased to eradicate the tumor cells. Before the childbirth we consider $u_{max} = 1$ mg L$^{-1}$ day$^{-1}$, and after the child birth we choose $u_{max} = 3.6$ mg L$^{-1}$ day$^{-1}$. Therefore, two DRL agents are trained before and after delivery. Fig. 5 shows the results of chemotherapy for this patient. It can be seen that in the initial 20 days, the drug concentration in the plasma is limited to 1 mg L$^{-1}$ day$^{-1}$. After delivery, the drug dose is increased to a limit of 3.6 mg L$^{-1}$ day$^{-1}$ to complete the treatment period. The plot of the drug administrated for this case is shown in Fig. 6. The method of Padmanabhan et al. exposes a similar behavior, while again having a longer medication period as observed in both Figs. 4 and 5.

Case 3: An old patient

In the third case, an elderly patient is considered who has cancer along with other critical illnesses. In this case, preserving a significant number of normal cells is necessary while eliminating the tumor cells.

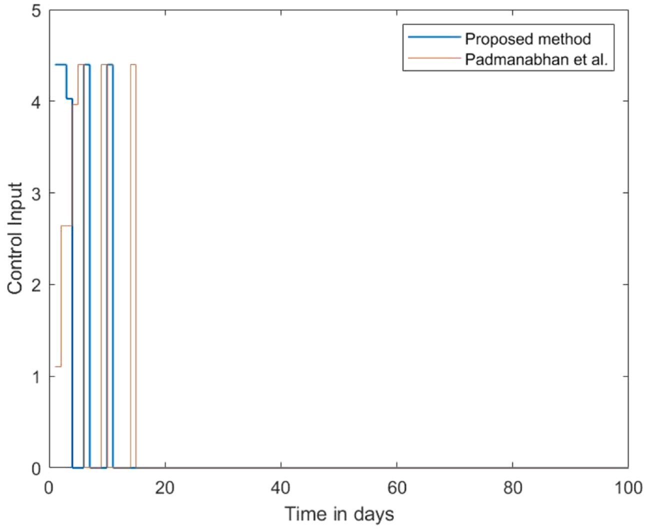


**Fig. 8.** Amount of drug administrated (Case 1) $\boldsymbol{u_{max}} = 4.4\ \boldsymbol{mgl^{-1}day^{-1}}$.

Therefore, other than attempting to reach the desired $x_{2d}$=0, the value of $x_1(t)$ should reach 1 ($x_{1d}$=1). Hence, we use a combination of the two states to define $e(t) = \delta x_2(t) + (1-\delta)(1-x_1(t))$. The parameter δ is used to weight the two elements of the error function and is set to 0.9 in the simulation. Figs. 6 and 7 show the response of the simulated old patient after the drug administration according to the DRL agent and Padmanabhan et al. [18]. The upper limit of the administrated drug is limited to 1.9 mg L$^{-1}$ day$^{-1}$ to reduce the damage to the normal cells. Both

**Table 2**
Comparison of the proposed method with the method of Padmanabhan et al. [18].

| | Proposed method | | | Padmanabhan et al. [18] | | |
|---|---|---|---|---|---|---|
| | Case 1 | Case 2 | Case 3 | Case 1 | Case 2 | Case 3 |
| Days to $x_2 = 10^{-4}$ | **46** | **62** | **54** | 50 | 66 | 55 |
| Integral of drug curve | **19.93** | 26.37 | **21.84** | 23.54 | **26.10** | 22.8 |

**Table 3**
Comparison of the proposed method with the methods presented in [58].

| Controller | total drug dosage |
|---|---|
| Synergetic control [58] | 24.000 |
| State feedback control [58] | 14.8637 |
| Fuzzy control Case I [58] | 27.2003 |
| Fuzzy control Case II [58] | 24.4707 |
| PID control [58] | 24.7328 |
| Proposed method | **19.9300** |

methods show a similar behavior, while the Q-learning-based algorithm of Padmanabhan et al. imposes a longer medication period. (Fig. 8)

As observed in the previous cases, the proposed method and the method of Padmanabhan et al. have a similar trend from different aspects of the system. Nevertheless, the latter Q-learning-based method, which requires discretization of both the states and actions of the learning agent, imposes a longer mediation period to eradicate the tumor cells. This is further illustrated in Table 2, where the number of days to reach the state of nearly eliminating the tumor cells ($x_2 \leq 1e - 4$) is shown. Another interesting factor is the total amount of the drug administrated by the two methods. The integrals of the control input curves for the three cases are also shown in Table 2. The results of our algorithm are averaged over three different training processes. The method of Padmanabhan et al. has a longer drug injection period and larger curve integrals in Cases 1 and 3. For the pregnant patient (case 2), it has a slightly smaller curve integral, while the drug administration period is still longer. Despite this, it is not able to eliminate the tumor cells faster. We performed a one-sample $t$-test and observed a significant difference between the reported results of our algorithm and the baseline algorithm of Padmanabhan et al. (for DF=2, $p < 0.001$ in case 1, $p < 0.05$ for days of case 2, and $p < 0.05$ for the old patient). The results show that discretization of the actions and states for reinforcement learning can increase the total amount of the administrated drug, while it also increases the training process of the algorithm and imposes a longer treatment period. The method proposed in this paper, which handles the state and action in their original continuous spaces, reduces the treatment period, the total amount of drug administrated, and the time to reach the goal state of eliminating the tumor cells.

To compare the results with non-RL state-of-the-art methods, the total drug dosage is compared with the strategies presented in [58]. In [58], five control strategies have been proposed for the 4-state tumor-immune interaction model, i.e. synergetic control, state feedback control, fuzzy control case I, fuzzy control case II, and PID control. The convergence time and total drug dosage of these strategies have been compared. We can only compare the total drug dosage used for treatment because the behavior of the state of the systems and the convergence time depend on the maximum admissible dose which is different from the present work. As shown in Table 3, the total drug dosage in our work is smaller except for the state feedback control strategy. However, it has to be noted that the state feedback control

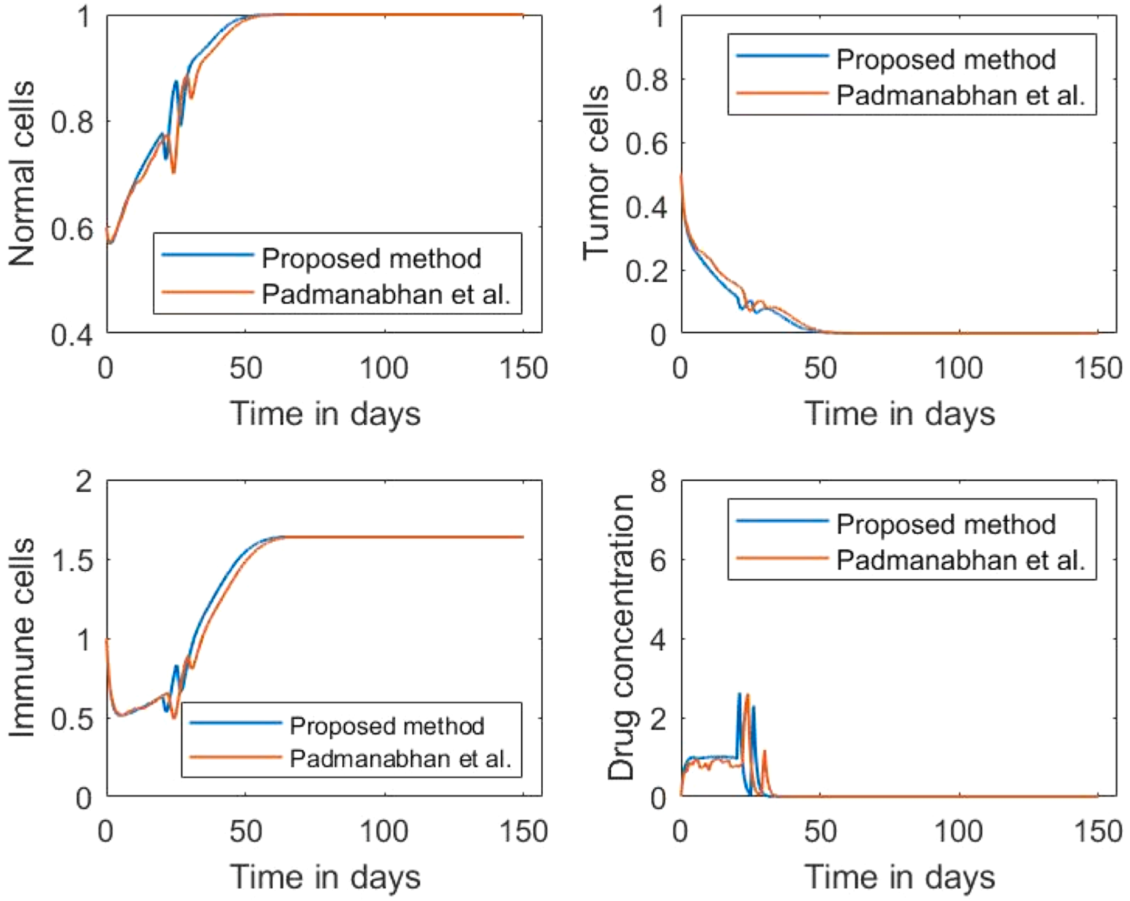


**Fig. 9.** Response of young pregnant woman with cancer (Case 2), $u_{max} = 1\ mgl^{-1}day^{-1}$until delivery (20 days) and then $u_{max} = 3.6\ mgl^{-1}day^{-1}$.

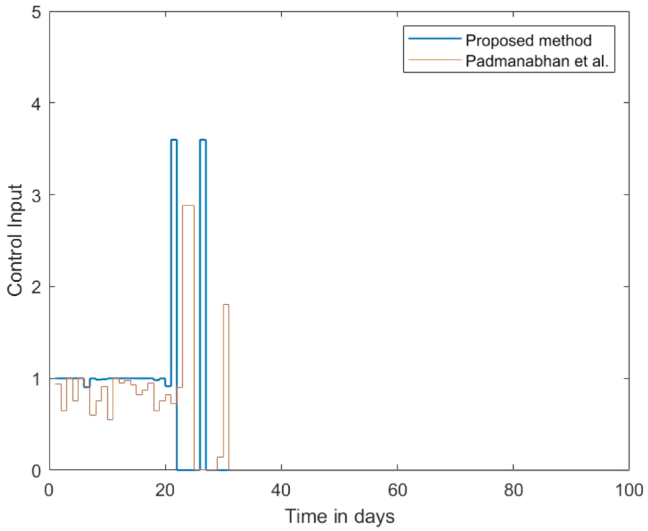


**Fig. 10.** Amount of drug administrated (Case 2), $u_{max} = 1\ mgl^{-1}day^{-1}$ until delivery (20 days) and then $u_{max} = 3.6\ mgl^{-1}day^{-1}$.

strategy, in contrast to other strategies, is a model-based strategy.

### 3.3. Model training convergence

The learning curves of the proposed method in terms of the error rate for the young and old patients (cases 1 and 3), are shown in Fig. 9. The error rate is averaged over different episodes for 30,000 training steps, and the standard deviation of the values is shown by the red-shaded area. The jitter of the blue curve is caused by the environmental change in each step. It can be seen that, in the training process, the trend of the error rate ($x_2$) is descending for both cases. The convergence rate of the old patient is slower compared to the young patient, due to the more complex constraints imposed in this case. The distribution of the error values demonstrated that the convergence rate improves as number of the training episodes increases.

### 3.4. Robustness of the controller

To show the robustness of the proposed DRL-based controller, three cases are considered. Case (i) has the nominal model with parameters presented in Table 1. In case (ii) and case (iii), the parameters of the model are changed with ±10 % parameters variation in nominal parameters presented in Table 1. Fig. 10 shows the behavior of these three cases by using the trained optimal DRL-based controller for the nominal model. It can be seen that the proposed controller can remove cancer cells despite to change in the parameters of the system since the decision of the controller is made based on the optimal policy concerning the state $s_k$, and the error $e(t)$. The control inputs using the trained DRL-based controller for the three cases are shown in Fig. 11.

### 3.5. Limitations

To improve the results, different extensions of the current study may be considered. In this study, a four-state mathematical model is considered as the tumor-immune interaction environment. Although this model was experimentally validated and used in many previous works, it may have some deviation from the real-world conditions. Due to this limitation, the robustness study has been conducted in Section

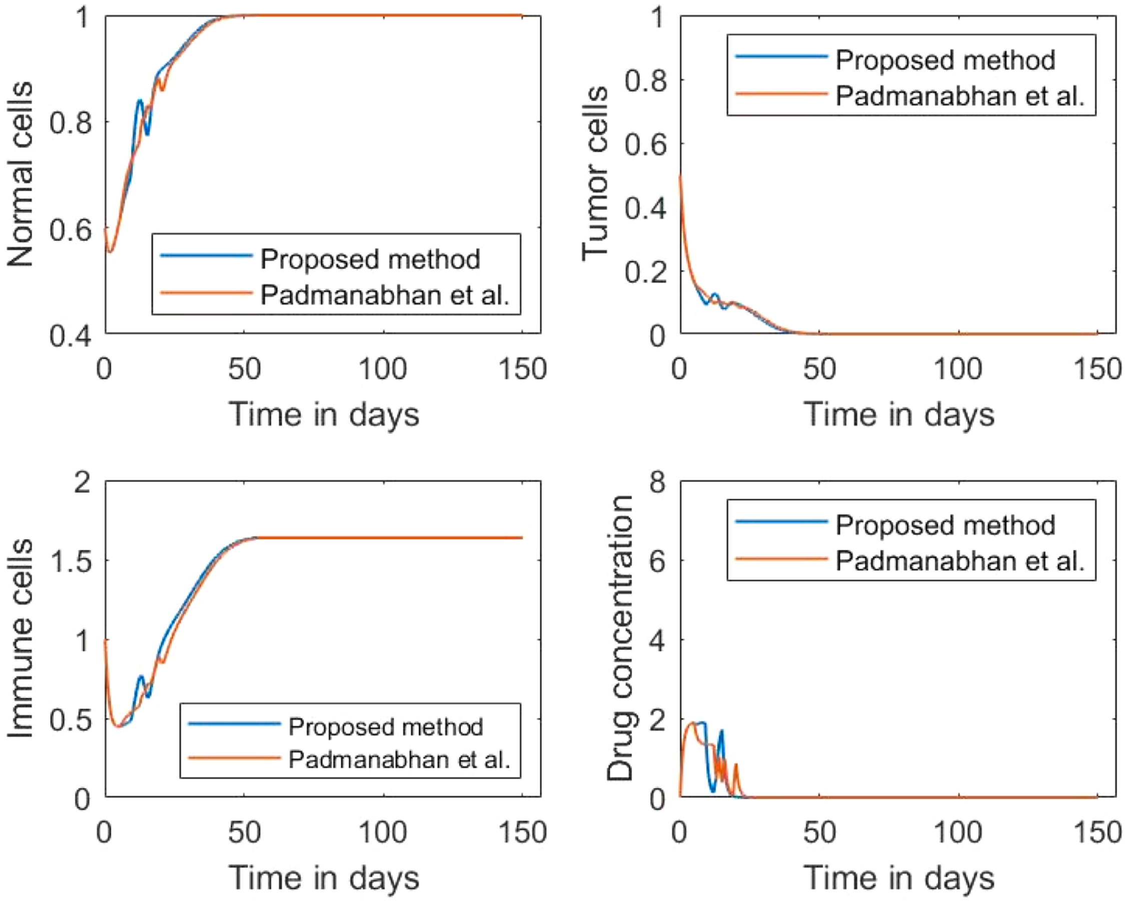


**Fig. 11.** Response of an elderly patient who has cancer along with other critical illnesses (Case 3), $u_{max} = 1.9\ mgl^{-1}day^{-1}$.

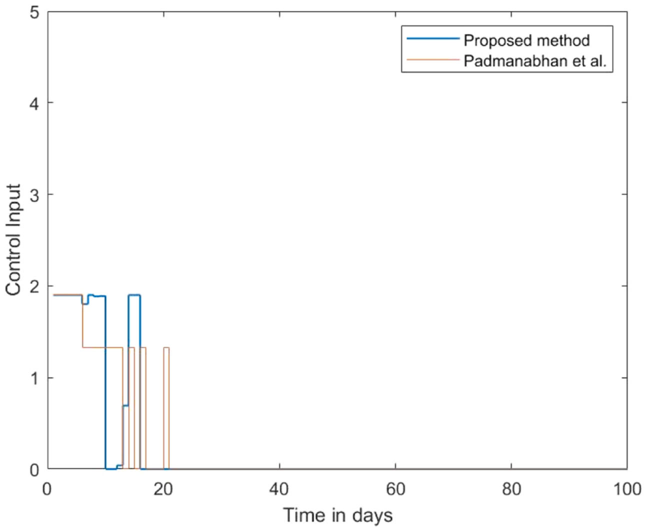


**Fig. 12.** Amount of drug administrated (Case 3), $u_{max} = 1.9\ mgl^{-1}day^{-1}$.

3.4. However, considering mathematical models with more cell interactions is a future direction that may enhance the results. Another challenge in real-world applications is the need to make sure the neural models have been trained for different possible conditions. The proposed model works in continuous state and action spaces, which is more realistic, and requires a significant number of training examples to generalize for patients with diverse conditions. In such cases, choosing the appropriate reward function is also of great importance. (Figs. 12, 13, 14, 15)

## 4. Discussion

The administration of an appropriate dosing schedule to effectively eliminate the cancer cells while protecting the patient's safety is a major concern in chemotherapy. The experimental results revealed that DRL can be considered as an adaptive control technique to administrate the drug dosing in chemotherapy. DRL is capable of taking a particular targeted adaption, where an agent can learn from its experiences and increase its overall reward while trying to achieve a certain goal efficiently [13,59,60]. In the present work, the DRL technique was used as a closed-loop optimal control problem. In other words, a DRL-based controller was proposed.

A closed-loop model-free controller using the Q-learning algorithm was presented in [18,61]. However, in [18,61], they proposed an RL-based controller by discretizing the state and action spaces, while in this paper a DRL-based controller has been proposed by considering continuous state and action spaces. The comparison with [18] shows the superiority of the proposed DRL-based controller.

Similar to [61], a controller based on reinforcement learning was presented in [62] using a rough estimate of the tumor size and the overall patient based on a TS model. They split the reward function into three distinct parts. A four-state model was used in [18] to derive the chemo-drug dose using a closed-loop RL-based controller. Using the TS model and the CC model in implementing an RL-based controller was compared in [59]. The cancer model considered in this paper is as same as [18], which was validated by experimental tests. Moreover, we considered three diverse sets of patients and exposed the capability of the deep RL method in automatically administrating the chemotherapy drug dose for adaptive control of the disease. The reward function for the RL agent is defined based on an error value. This error value depends on the number of tumor cells and also the number of normal cells if the immune system is weak. The results showed that this reward function is capable of guiding the agent in selecting appropriate actions toward the goal state. We evaluated the proposed method in terms of variations in different variables and observed that in all different cases, the algorithm is effective in the treatment of the disease. As the baseline, we compared our work with the method of Padmanabhan et al. [18], which uses the Q-learning method for drug dose administration in discrete state and action spaces. It was observed that the deep RL method which operates in the original finite space has a better performance in both treatment time and the total amount of administrated drug. The proposed method also reduces the requirement of expert interference to determine the discretization rules. We also conducted a robustness experiment to show that the model is effective despite changes in the parameters of the system.

## 5. Conclusion

In this paper, the problem of cancer chemotherapy control as an optimization problem, providing a solution based on deep reinforcement learning was considered. A patient with a nonlinear pharmacological cancer model was simulated. The reinforcement learning framework enabled a model-free approach to drug dosing control. By designing a deep RL controller, the system variables in their original continuous spaces were modeled. This is in contrast to the available model-free approaches, which commonly discretize variables. Therefore, the real-

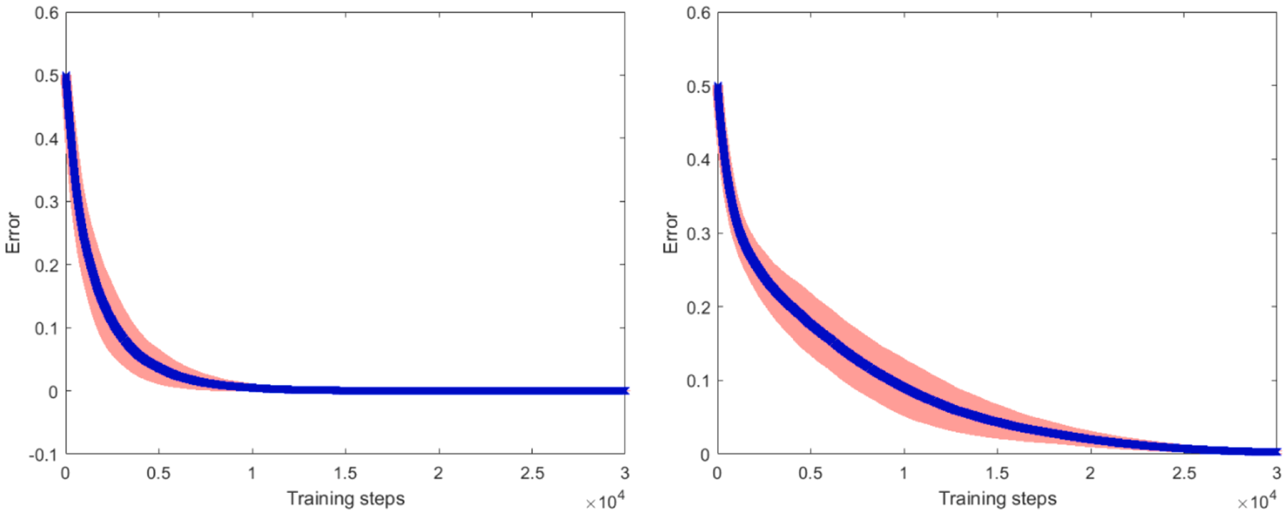


**Fig. 13.** The convergence of training the DRL control model.

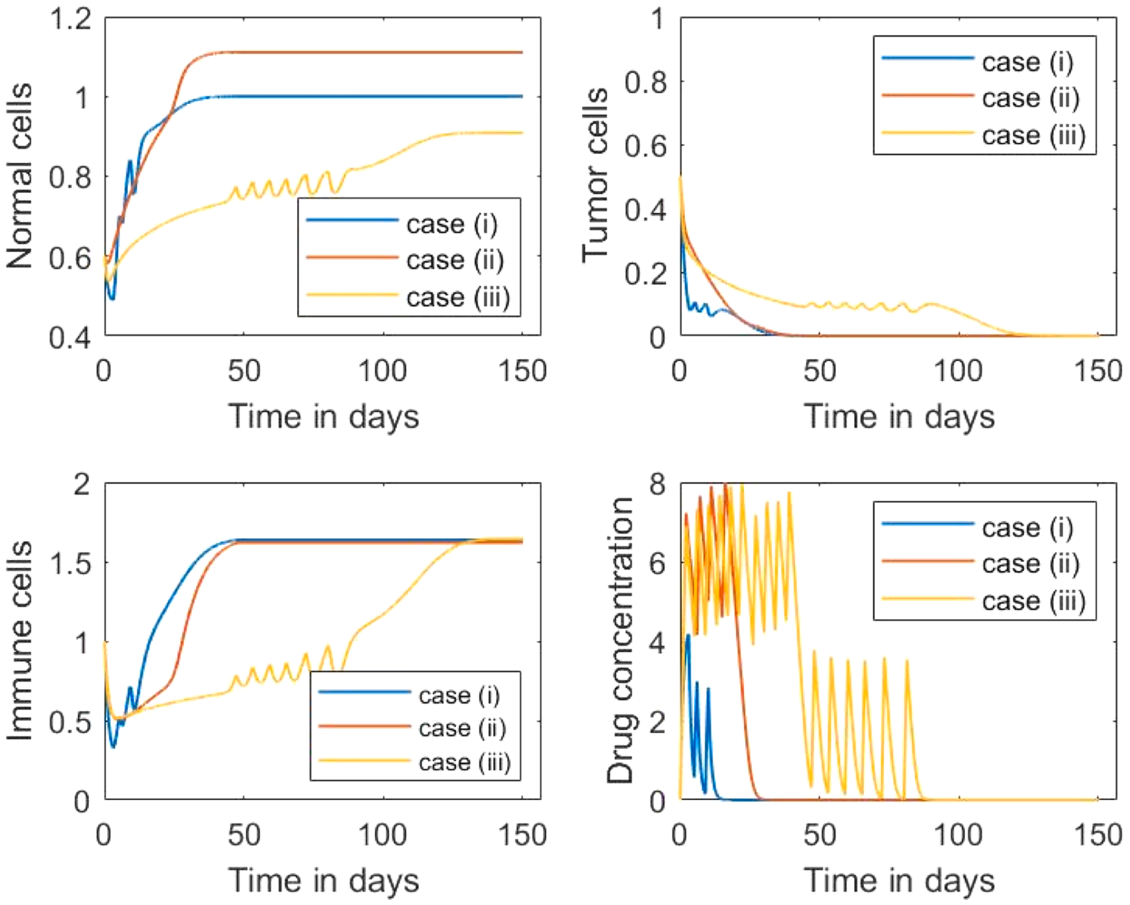


**Fig. 14.** Response for three different patients using trained DRL-based controller; Case (i) with nominal model, Case (ii) −10 % parameter variation, Case (iii) with +10 % parameter variation.

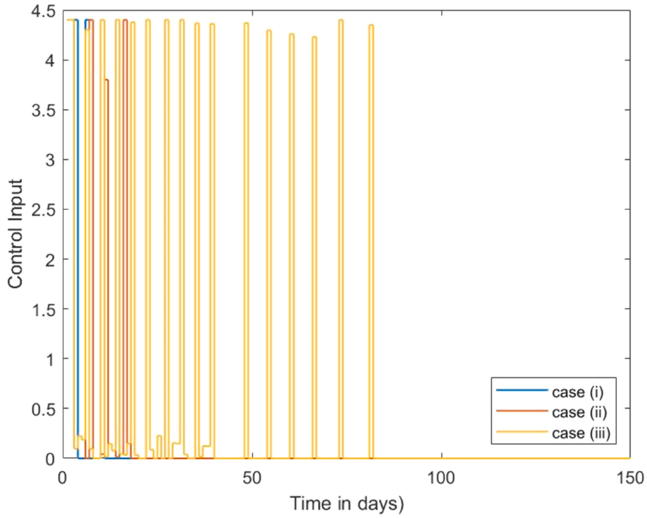


**Fig. 15.** Control input for three different patient models; Case (i) with nominal model, Case (ii) with −10 % parameter variation, Case (iii) with +10 % parameter variation.

world conditions can be simulated more accurately, and reduce the expert intervention. A diverse set of experiments to expose the performance of the proposed model from different perspectives was conducted. Results on three different patients and comparison with previous work show the efficiency of the proposed model in controlling the cancer treatment process. The actual application of DRL to real-life medical cases is still in an emerging state. Different challenges should be addressed in this regard, such as dynamic patient regimens and personalized medicine. Such issues require both extended theoretical and practical analysis.

## Ethical approval

No ethics approval was required.

## Data availability

Not applicable.

## Declaration of Competing Interest

The authors declare that they have no conflicts of interest.

## Funding

No funding was received for this study.

## References

[1] American Cancer Society, Cancer facts & figures 2022. Atlanta, Ga, Am. Cancer Soc. (2022) 1–76. https://www.cancer.org/cancer/bladder-cancer/detection-diagnosis-staging/survival-rates.html.
[2] R. Elancheran, V.L. Maruthanila, M. Kumar, J. Kotoky, S. Kabilan, Strategy towards diagnosis and treatment for prostate cancer, Urol. Res. Ther. J. 1 (2017) 115.
[3] P. Yazdjerdi, N. Meskin, M. Al-Naemi, A.E. Al Moustafa, L. Kovács, Reinforcement learning-based control of tumor growth under anti-angiogenic therapy, Comput.

Methods Programs Biomed. 173 (2019) 15–26, https://doi.org/10.1016/j.cmpb.2019.03.004.

[4] V.L. Maruthanila, R. Elancheran, A.B. Kunnumakkara, S. Kabilan, J. Kotoky, Recent development of targeted approaches for the treatment of breast cancer, Breast Cancer 24 (2017) 191–219, https://doi.org/10.1007/s12282-016-0732-1.

[5] B. Dhar, P.K. Gupta, Mathematical analysis on the behaviour of tumor cells in the presence of monoclonal antibodies drug, Smart Innov. Syst. Technol. 206 (2021) 311–321, https://doi.org/10.1007/978-981-15-9829-6_24.

[6] Y. Su, D. Sun, Optimal control of anti-HBV treatment based on combination of Traditional Chinese Medicine and Western Medicine, Biomed. Signal Process. Control. 15 (2015) 41–48, https://doi.org/10.1016/j.bspc.2014.09.007.

[7] M. Nazari, N. Babaei, M. Nazari, Nonlinear SDRE based adaptive fuzzy control approach for age-specific drug delivery in mixed chemotherapy and immunotherapy, Biomed. Signal Process. Control. (2021) 68, https://doi.org/10.1016/j.bspc.2021.102687.

[8] L. dePillis, Mathematical model of colorectal cancer with monoclonal antibody treatments, Br. J. Med. Med. Res. 4 (2014) 3101–3131, https://doi.org/10.9734/bjmmr/2014/8393.

[9] Y.P. Liu, C.C. Zheng, Y.N. Huang, M.L. He, W.W. Xu, B. Li, Molecular mechanisms of chemo- and radiotherapy resistance and the potential implications for cancer treatment, MedComm 2 (2021) 315–340, https://doi.org/10.1002/mco2.55.

[10] A. Ghaffari, B. Bahmaie, M. Nazari, A mixed radiotherapy and chemotherapy model for treatment of cancer with metastasis, Math. Methods Appl. Sci. 39 (2016) 4603–4617, https://doi.org/10.1002/mma.3887.

[11] M. Sharifi, H. Moradi, Nonlinear composite adaptive control of cancer chemotherapy with online identification of uncertain parameters, Biomed. Signal Process. Control. 49 (2019) 360–374, https://doi.org/10.1016/j.bspc.2018.07.009.

[12] T. Chen, N.F. Kirkby, R. Jena, Optimal dosing of cancer chemotherapy using model predictive control and moving horizon state/parameter estimation, Comput. Methods Programs Biomed. 108 (2012) 973–983, https://doi.org/10.1016/j.cmpb.2012.05.011.

[13] C.Y. Yang, C. Shiranthika, C.Y. Wang, K.W. Chen, S. Sumathipala, Reinforcement learning strategies in cancer chemotherapy treatments: a review, Comput. Methods Programs Biomed. 229 (2023), https://doi.org/10.1016/j.cmpb.2022.107280.

[14] H. Sbeity, Review of optimization methods for cancer chemotherapy treatment planning, J. Comput. Sci. Syst. Biol. 8 (2015), https://doi.org/10.4172/jcsb.1000173.

[15] O. Shindi, J. Kanesan, G. Kendall, A. Ramanathan, The combined effect of optimal control and swarm intelligence on optimization of cancer chemotherapy, Comput. Methods Programs Biomed. 189 (2020), https://doi.org/10.1016/j.cmpb.2020.105327.

[16] R. Padmanabhan, N. Meskin, A.-E. Al Moustafa, Mathematical models of cancer and different therapies, (2021). 10.1007/978-981-15-8640-8.

[17] N. Darandis, M. Nazari, A new mathematical modeling and sub-optimal chemotherapy of cancer, J. Biol. Syst. 29 (2021) 647–685, https://doi.org/10.1142/S0218339021500133.

[18] R. Padmanabhan, N. Meskin, W.M. Haddad, Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment, Math. Biosci. 293 (2017) 11–20, https://doi.org/10.1016/j.mbs.2017.08.004.

[19] K.L. Kiran, D. Jayachandran, S. Lakshminarayanan, Multi-objective optimization of cancer immuno-chemotherapy, IFMBE Proc. 23 (2009) 1337–1340, https://doi.org/10.1007/978-3-540-92841-6_329.

[20] S.L. Noble, E. Sherer, R.E. Hannemann, D. Ramkrishna, T. Vik, A.E. Rundell, Using adaptive model predictive control to customize maintenance therapy chemotherapeutic dosing for childhood acute lymphoblastic leukemia, J. Theor. Biol. 264 (2010) 990–1002, https://doi.org/10.1016/j.jtbi.2010.01.031.

[21] M. Engelhart, D. Lebiedz, S. Sager, Optimal control for selected cancer chemotherapy ODE models: a view on the potential of optimal schedules and choice of objective function, Math. Biosci. 229 (2011) 123–134, https://doi.org/10.1016/j.mbs.2010.11.007.

[22] A. Ghaffari, M. Nazari, F. Arab, Suboptimal mixed vaccine and chemotherapy in finite duration cancer treatment: state-dependent Riccati equation control, J. Brazilian Soc. Mech. Sci. Eng. 37 (2015) 45–56, https://doi.org/10.1007/s40430-014-0172-9.

[23] M. Nazari, A. Ghaffari, F. Arab, Finite duration treatment of cancer by using vaccine therapy and optimal chemotherapy: state-dependent Riccati equation control and extended Kalman filter, J. Biol. Syst. 23 (2015) 1–29, https://doi.org/10.1142/S0218339015500011.

[24] K.C. Tan, T.H. Lee, J. Cai, Y.H. Chew, Automating the drug scheduling of cancer chemotherapy via evolutionary computation, in: Proc. 2002 Congr. Evol. Comput. CEC 2002 1, 2002, pp. 908–913, https://doi.org/10.1109/CEC.2002.1007046.

[25] D. Vrabie, K.G. Vamvoudakis, F.L. Lewis, Optimal adaptive control and differential games by reinforcement learning principles, Optim. Adapt. Control Differ. Games by Reinf. Learn. Princ. (2012) 1–289, https://doi.org/10.2514/1.g000173.

[26] R.S. Sutton, A.G. Barto, Reinforcement learning: an introduction, IEEE Trans. Neural Netw. 9 (1998), https://doi.org/10.1109/tnn.1998.712192, 1054–1054.

[27] R. Padmanabhan, N. Meskin, W.M. Haddad, Reinforcement learning-based control of drug dosing with applications to anesthesia and cancer therapy, Control Appl. Biomed. Eng. Syst. (2020) 251–297, https://doi.org/10.1016/B978-0-12-817461-6.00009-3.

[28] G. Yauney, P. Shah, Reinforcement learning with action-derived rewards for chemotherapy and clinical trial dosing regimen selection, Proc. Mach. Learn. Res. 85 (2018) 161–226.

[29] S. Ebrahimi, G.J. Lim, A reinforcement learning approach for finding optimal policy of adaptive radiation therapy considering uncertain tumor biological response, Artif. Intell. Med. (2021) 121, https://doi.org/10.1016/j.artmed.2021.102193.

[30] M.K. Bothe, L. Dickens, K. Reichel, A. Tellmann, B. Ellger, M. Westphal, A.A. Faisal, The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas, Expert Rev. Med. Devices. 10 (2013) 661–673, https://doi.org/10.1586/17434440.2013.827515.

[31] T. Li, Z. Wang, W. Lu, Q. Zhang, D. Li, Electronic health records based reinforcement learning for treatment optimizing, Inf. Syst. 104 (2022), https://doi.org/10.1016/j.is.2021.101878.

[32] T. Degris, P.M. Pilarski, R.S. Sutton, Model-free reinforcement learning with continuous action in practice, Proc. Am. Control Conf. (2012) 2177–2182, https://doi.org/10.1109/acc.2012.6315022.

[33] R.K. Tan, Y. Liu, L. Xie, Reinforcement learning for systems pharmacology-oriented and personalized drug design, Expert Opin. Drug Discov. 17 (2022) 849–863, https://doi.org/10.1080/17460441.2022.2072288.

[34] J.D. Martín-Guerrero, F. Gomez, E. Soria-Olivas, J. Schmidhuber, M. Climente-Martí, N.V. Jiménez-Torres, A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients, Expert Syst. Appl. 36 (2009) 9737–9742, https://doi.org/10.1016/j.eswa.2009.02.041.

[35] B.L. Moore, L.D. Pyeatt, V. Kulkarni, P. Panousis, K. Padrez, A.G. Doufas, Reinforcement learning for closed-loop propofol anesthesia: a study in human volunteers, J. Mach. Learn. Res. 15 (2014) 655–696.

[36] R. Padmanabhan, N. Meskin, W.M. Haddad, Optimal adaptive control of drug dosing using integral reinforcement learning, Math. Biosci. 309 (2019) 131–142, https://doi.org/10.1016/j.mbs.2019.01.012.

[37] V. Konda, J. Tsitsiklis, Actor-critic algorithms, in: S. Solla, T. Leen, K. Müller (Eds.), Adv. Neural Inf. Process. Syst., MIT Press, 2000.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533, https://doi.org/10.1038/nature14236.

[39] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, in: 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc, 2016.

[40] I.G, Y.B, A. Courville, Deep learning 简介 一 , 什么是 deep learning ∝, Nature 29 (2016) 1–73, https://doi.org/10.3902/jnns.21.192.

[41] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-Learning, in: 30th AAAI Conf. Artif. Intell. AAAI 2016, 2016, pp. 2094–2100, https://doi.org/10.1609/aaai.v30i1.10295.

[42] Q. Cai, L. Pan, P. Tang, Deterministic policy gradients with general state transitions, (2018). http://arxiv.org/abs/1807.03708.

[43] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, CoRR (2015).

[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, ArXiv Prep. (2013). http://arxiv.org/abs/1312.5602.

[45] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: 32nd Int. Conf. Mach. Learn. ICML 2015 1, 2015, pp. 448–456.

[46] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, Z. Yi, Deep-reinforcement-learning-based autonomous voltage control for power grid operations, IEEE Trans. Power Syst. 35 (2020) 814–817, https://doi.org/10.1109/TPWRS.2019.2941134.

[47] Z. Ning, K. Zhang, X. Wang, M.S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, R.Y.K. Kwok, Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning-based traffic control system, IEEE Trans. Intell. Transp. Syst. 22 (2021) 5201–5212, https://doi.org/10.1109/TITS.2020.2970276.

[48] T. Zhu, K. Li, P. Herrero, P. Georgiou, Basal glucose control in type 1 diabetes using deep reinforcement learning: an in silico validation, IEEE J. Biomed. Heal. Inf. 25 (2021) 1223–1232, https://doi.org/10.1109/JBHI.2020.3014556.

[49] I. Fox, J. Lee, R. Pop-Busui, J. Wiens, Deep reinforcement learning for closed-loop blood glucose control, (2020). http://arxiv.org/abs/2009.09051.

[50] M. Tortora, E. Cordelli, R. Sicilia, M. Miele, P. Matteucci, G. Iannello, S. Ramella, P. Soda, Deep reinforcement learning for fractionated radiotherapy in non-small cell lung carcinoma, Artif. Intell. Med. 119 (2021), https://doi.org/10.1016/j.artmed.2021.102137.

[51] L. Huo, Y. Tang, Multi-objective deep reinforcement learning for personalized dose optimization based on multi-indicator experience replay, Appl. Sci. (2023) 13, https://doi.org/10.3390/app13010325.

[52] L.G. De Pillis, A. Radunskaya, A mathematical tumor model with immune resistance and drug therapy: an optimal control approach, J. Theor. Med. 3 (2001) 79–100, https://doi.org/10.1080/10273660108833067.

[53] L.G. De Pillis, A. Radunskaya, The dynamics of an optimally controlled tumor model: a case study, Math. Comput. Model. 37 (2003) 1221–1244, https://doi.org/10.1016/S0895-7177(03)00133-X.

[54] A. Talkington, C. Dantoin, R. Durrett, Ordinary differential equation models for adoptive immunotherapy, Bull. Math. Biol. 80 (2018) 1059–1083, https://doi.org/10.1007/s11538-017-0263-8.

[55] H. Hasselt, Double Q-learning, in: J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (Eds.), Adv. Neural Inf. Process. Syst., Curran Associates, Inc., 2010.

[56] S. Fujimoto, H. Van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, 35th Int. Conf. Mach. Learn. ICML 4 (2018) 2587–2601, 2018.

[57] M. Nazari, A. Ghaffari, The effect of finite duration inputs on the dynamics of a system: proposing a new approach for cancer treatment, Int. J. Biomath. 8 (2015), https://doi.org/10.1142/S1793524515500369.

[58] H. Qaiser, I. Ahmad, M. Kashif, Fuzzy, synergetic and non-linear state feedback control of chemotherapy drug for a cancerous tumor, Biomed. Signal Process. Control. 62 (2020), https://doi.org/10.1016/j.bspc.2020.102061.

[59] C. Shiranthika, K.W. Chen, C.Y. Wang, C.Y. Yang, B.H. Sudantha, W.F. Li, Supervised optimal chemotherapy regimen based on offline reinforcement learning, IEEE J. Biomed. Heal. Inf. 26 (2022) 4763–4772, https://doi.org/10.1109/JBHI.2022.3183854.

[60] B. Eastman, M. Przedborski, M. Kohandel, Reinforcement learning derived chemotherapeutic schedules for robust patient-specific therapy, Sci. Rep. 11 (2021), https://doi.org/10.1038/s41598-021-97028-6.

[61] R. Padmanabhan, N. Meskin, W.M. Haddad, Learning-based control of cancer chemotherapy treatment, 50 (2017) 15127–15132. https://doi.org/10.1016/j.ifacol.2017.08.2247.

[62] Y. Zhao, M.R. Kosorok, D. Zeng, Reinforcement learning design for cancer clinical trials, Stat. Med. 28 (2009) 3294–3315, https://doi.org/10.1002/sim.3720.