Computers in Biology and Medicine 148 (2022) 105860

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Subcutaneous insulin administration by deep reinforcement learning for blood glucose level control of type-2 diabetic patients

Mohammad Ali Raheb [a], Vahid Reza Niazmand [b], Navid Eqra [c,*], Ramin Vatankhah [a]

[a] School of Mechanical Engineering, Shiraz University, Shiraz, Iran
[b] Department of IT and Computer Engineering, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
[c] Department of Mechanical Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Type-2 diabetes
Insulin administration
Subcutaneous injection
Normalized advantage function

ABSTRACT

Background: Type-2 diabetes mellitus is characterized by insulin resistance and impaired insulin secretion in the human body. Many endeavors have been made in terms of controlling and reducing blood glucose via the medium of automated controlling tools to increase precision and efficiency and reduce human error. Recently, reinforcement learning algorithms are proved to be powerful in the field of intelligent control, which was the motivation for the current study.

Methods: For the first time, a reinforcement algorithm called normalized advantage function (NAF) algorithm has been applied as a model-free reinforcement learning method to regulate the blood glucose level of type-2 diabetic patients through subcutaneous injection. The algorithm has been designed and developed in a model-free approach to avoid additional inaccuracies and parameter uncertainty introduced by the mathematical models of the glucoregulatory system. Insulin doses constitute the control action that is designed to be stated directly in clinical language with the unit *IU*. In this regard, a new environment state is considered in addition to the glucose level to take into account the delayed effect of insulin elimination under the skin. Finally, a simple but practical reward function is developed to be used with the NAF algorithm to correct the glucose level and maintain it in the desired range.

Results: The simulation environment was set up to imitate the basal-bolus process accurately. Results for 30 days of simulation of the designed controller on three different average virtual patients verify the feasibility and effectiveness of the method and reveal our proposed controller's learning ability. Moreover, as the insulin elimination dynamic was taken into account, a more complete and more realistic model than the previously studied models has emerged.

Conclusion: NAF has proved a promising control approach, able to successfully regulate and significantly reduce the fluctuation of the blood glucose without meal announcements, compared to standard optimized open-loop basal-bolus therapies. The method and its results, which are directly in the clinical language, are applicable in real-time clinical situations.

1. Introduction

Diabetes Mellitus (DM) is one of the most prevalent chronic diseases that modern man has to contend with. Common metabolic disorders contributing to such a disease comprise reduced insulin secretion, decreased glucose utilization, and increased glucose production. On the basis of the pathogenic process leading to hyperglycemia, DM gets classified into two predominant categories designated as type-1 and type-2. Type-1 DM develops as a result of autoimmunity against insulin-producing beta cells, causing complete or near-total insulin deficiency. On the other hand, type-2 DM, which is more common than the former type, is characterized by insulin resistance and impaired insulin secretion [1]. Patients diagnosed with type-2 DM need insulin to take their condition under control. As the disease has pernicious effects on organs such as the eye and heart, it is paramount to control the condition using an effective mechanism. DM is often self-managed by the patients through multiple glucose level measurements on a daily basis and administration of insulin via injection or a pump, which can become quite an ordeal [1]. Many endeavors have been made in the past recent years in terms of controlling and reducing blood glucose via the medium

* Corresponding author.
E-mail address: navid.eqra@sharif.edu (N. Eqra).

https://doi.org/10.1016/j.compbiomed.2022.105860
Received 30 December 2021; Received in revised form 22 June 2022; Accepted 26 June 2022
Available online 14 July 2022

of automated controlling tools in order to increase precision and reduce human error as much as possible. However, this study investigates the utilization of an artificial intelligence algorithm called normalized advantage functions (NAF). The NAF algorithm is applied as an RL method in order to identify the patient's condition and help control the blood glucose level in an optimal way by subcutaneous injection. First, a dynamic model of the problem is introduced, used for training the RL agent. The model includes the absorption dynamics of the injected insulin in addition to the glucose-insulin dynamic system. This gives a more complete and more realistic model than the previously studied models for designing the controller. It also helps to state the study's outcome directly in clinical language. More specifically, the contributions of this study can be summarized as follows:

1 Through considering a tradeoff between insulin dosage and desirable insulin and glucose level, a robust adaptive treatment framework based on model-free reinforcement learning (RL) algorithm is represented to regulate the blood glucose level of type two diabetic patients through subcutaneous injection. The model-free nature of this approach lets it function without any expert knowledge of the underlying dynamics or meal announcements, using only glucose level as its input.
2 A simple and practical reward function is introduced to be used with the algorithm in order to correct the glucose level and preserve it in the desired range.
3 The proposed algorithm's performance is compared to two other approaches used to administrate dynamic treatment regimens. Unlike similar studies [2,3], the system is formulated to make a good candidate for practical use, considering all the limitations and challenges in real-life conditions. Thus, the absorption dynamic of the injected insulin is taken into account in the environment set up in addition to the regular basal-bolus process. Also, the algorithm is designed to yield the insulin dose levels in the functional unit *IU*, commonly used by practitioners in the field.

The remainder of this paper is organized as follows: Section 2 is dedicated to an overview of the present approaches and a more in-depth introduction of RL and NAF, along with their merits for using them in this case. In Section 3, the dynamic models are defined. Then, Section 4 is dedicated to the problem formulation and elaborating our RL solution, and the performance of the algorithm is evaluated on three different virtual patients in Section 5. Lastly, the results of the experiments are illustrated and discussed in Section 6, and conclusions are made in Section 7.

2. Related work

Computer algorithms have been broadly used as automated solutions to complex and sensitive problems with numerous contributing factors. For instance, Wu et al. investigated the usage of dummy query building algorithms to prevent exposure of e-commerce platform users' privacy against untrusted servers [4]. A similar effort has been made for location-based services in Ref. [5]. Both of these studies demonstrated promising results in the presence of users' privacy model without tampering with user experience. In another work, Yan et al. proposed a new algorithm for shapelet extraction for early classification on time series, which was able to acquire a classification accuracy comparable to full-time series while making earlier predictions [6]. The performance was tested based on 16 datasets of time series data, including electrocardiogram signals. As a huge subcategory of algorithms, artificial intelligence and machine learning (ML) algorithms are deemed great candidates to be used in healthcare applications formulated as optimization problems [7–9]. For instance, Wang et al. leveraged deep convolutional neural networks for lesion segmentation of endoscopy images, which could effectively utilize global semantic information and low-level images to produce high-resolution lesion segmentation [10]. In the field of DM, works can be divided into two categories: 1. Diabetes prediction, 2. Glucose management and control. In the first category, machine learning algorithms are broadly applied which their performance comparisons can be found in Ref. [11]. As the current study's focus is on the second category, further reviews are related to the studies on glucose control.

The performance of proportional-integral-derivative (PID), model predictive control (MPC), and fuzzy logic controller (FLC) [12–15] have been broadly investigated as methods of creating an artificial pancreas (AP), with PID being the most frequently used [13]. The simplicity of PID makes it easy to use, and in practice, it achieves strong results [13]. In contrast, the main drawback of a PID controller in the setting of a glucose controller is its reactivity. Since it only responds to current glucose values, it cannot respond fast enough to meals in order to sufficiently control postprandial excursions without meal announcements [16]. That is, it can overcorrect for these spikes without further supervision, triggering postprandial hypoglycemia [17]. Conversely, we argue that modern learning approaches, especially reinforcement learning (RL), can be a good substitute for conventional methods such as PID and are also well-suited for controlling blood glucose for the following reasons: 1. RL is a semi-supervised learning algorithm meaning that it does not need prior expert knowledge or meal announcements. It can extract and leverage related patterns such as regular meal times from the given system state using pattern recognition; 2. RL works with minimum assumptions about the structure of the underlying process, allowing the same system to adapt to different patients or changes in their condition over time without retraining; 3. Unlike conventional controllers, RL does not consider each state completely novel and uses past states with the new state to optimize the system. Furthermore, RL is an online learning algorithm and has the ability to learn while acting, making it a proper candidate to be used as a control system [18,19].

The most commonly used RL algorithms for blood glucose control in type-1 DM are actor-critic (AC), Q-Learning, SARSA, and Gaussian Process Reinforcement Learning (GPRL). Dynamic programming (DP) has also been utilized to control type-2 [20]. A common challenge with such model-free algorithms is that sample complexity, and high-order simulator functions necessitate a huge amount of data and samples to be fed to the system. Although it is negligible in simulation, it can become an impediment when working with physical systems [21]. Overall, an algorithm that can bring the generality of model-based algorithms using real-time data would be of desire.

Applying the RL framework, one maps some observations in terms of states like current blood glucose level to an action like a dose of insulin to be injected to maximize some notion of a delayed reward. Researchers in Healthcare have started to explore RL as a solution to find a proper treatment for patients since it reframes the problem from a diagnosis-based to an action-based problem [22]. Cases in point could be the treatment of sepsis [22,23] and mechanical ventilation [24]. In addition, RL has been explored to provide contextual suggestions for behavioral modifications [25]. RL can provide us with a viable solution to deal with the following challenge since it is suitable for learning complex behaviors, which should adapt to frequently changing domains [26]. On the other hand, as mentioned before, some RL algorithms suffer from lack of reliable data to train an effective policy. However, unlike many other diseases, credible simulators exist to mimic the glucoregulatory system's dynamics [27]. These models have been used before for learning purposes. To be more specific, researchers have investigated the use of off-policy learning to discover control parameters in diabetes simulations [28]. For instance, Ngo et al. introduced a blood glucose controller based on reinforcement learning and a feedforward algorithm for type-1 diabetes using basal and bolus insulin. The given results show that the algorithm was able to stabilize blood glucose while reducing glucose undershoot and preventing hypoglycemia, despite the variations in the model's response caused by uncertainties. Also, compared to PID, the algorithm substantially reduced the peak of post-meal glucose level as well as the distance between the lowest blood glucose and the ideal

blood glucose level, known as the undershoot blood glucose [29].

In another work, Fox and Wiens tried to demonstrate how deep model-free RL can be used as an improvement over alternative approaches to control blood glucose [15]. This study uses deep Q network (DQN) to tackle the problem and shows how it was able to outperform baseline approaches, even without using the ground-truth state. Although the results are encouraging, the proposed model was not able to outperform PID in all settings.

When the problem at hand is continuous on state/action space, normalized advantage function (NAF) is a feasible choice as the RL algorithm, which is devised to rectify the noted limitation with real-world data. NAF has also been used in control tasks before and showed major advances in learning and execution speed and better performance compared to similar algorithms such as deep deterministic policy gradient (DDPG), both in simulation and real-world robotic tasks [30–32]. Moreover, NAF avoids the need for a second actor or policy function, as opposed to actor-critic (AC) algorithms, resulting in a simpler algorithm.

3. Glucose-insulin dynamics

A mathematical model of the blood glucose system is crucial for developing and testing an artificial pancreas [33]. This model can also be used as the simulation environment to train intelligent insulin injection controllers [13]. Currently, available models are built using experimentation and expert knowledge of physiological phenomena. Eqs. (1) and (2) define a blood glucose-insulin model borrowed from Ref. [34] as a system of delayed differential equations (DDE) in the state-space form with the initial conditions as $G(0) = G_b, I(0) = I_b$. They are used as the environment later to train our controller agent. In this model, the two state variables G, I represent blood glucose level in mM and insulin level in pM, respectively. Control action $v(t)$ representing insulin input is also added to the model which would be calculated by the controller agent.

$$\begin{cases} \dot{G}(t) = -K_{xgi}G(t)I(t) + \dfrac{T_{gh}}{V_G} \\ \dot{I}(t) = -K_{xi}I(t) + \dfrac{T_{iG_{max}}}{V_I}\varphi\big(G(t-\tau_g)\big) + \dfrac{v(t)}{V_I} \end{cases} \tag{1}$$

$$\varphi\big(G(t-\tau_g)\big) = \frac{\left(\frac{G(t-\tau)}{G^*}\right)^{\gamma}}{1+\left(\frac{G(t-\tau)}{G^*}\right)^{\gamma}} \tag{2}$$

The model's parameters are described in Appendix 1, and their values are present in Table 1. They are different for every patient, however, our trained controller agent, which is model-free, can deal with this parameter uncertainty and identify the system dynamics itself.

Table 1
Environment parameters for the experiments.

	Patient 1 [34]	Patient 2 [36]	Patient 3 [36]
$I_b(pM)$	59.85	24.04	27.82
$G_b(mM)$	8.85	8.78	8.96
$G^*(mM)$	9	9	9
$\tau_g(min)$	6.5	0	0
$V_G(L/kgBW)$	0.18	0.13	0.13
$V_I(L/kgBW)$	0.25	0.25	0.25*
γ	15.92	2.52	2.3
$T_{gh}(min^{-1}(mmol/kgBW))$	0.0023	0.0027	0.0025
$T_{iGmax}(min^{-1}(pmol/kgBW))$	1.685	0.75	1.4
$K_{xi}(min^{-1})$	0.038	0.06	0.1
$K_{xgi}(min^{-1}pM^{-1})$	3.15e-5	9.96e-5	7.45e-5

*In this case, V_I was originally set as 0.248 in Ref. [36]. However, in this study it is rounded to 0.25 since it is considered constant in unit conversions. Accordingly, T_{gh}, T_{igmax} are recalculated here using Eq. (3) and (4).

The unit of the insulin input (control action) $v(t)$ is $min^{-1}(pmol/kgBW)$ in Eq. (1). As the subcutaneous injection is considered, the insulin is firstly injected under the skin with the unit IU. There exists a first-order dynamic for the insulin elimination under the skin (equivalently, the insulin absorption into the blood), as shown in Fig. 1 (a) [35]. To reach the appropriate quantity for the insulin input in Eq. (1), the function illustrated in Fig. 1 (b) is introduced, which is obtained by making some alterations in the insulin elimination function drawn in Fig. 1 (a). This unit conversion process could be also found in Appendix 2. Such a modeling procedure is done to design the final control action in IU which is used directly in practical applications of medical science.

4. Reinforcement learning control

In this section, the problem is formalized according to the reinforcement learning literature and a NAF agent is designed to be used as the controller of the system. Accordingly, in the following sections, the environment and the other components of our reinforcement learning framework are defined and formulated.

4.1. Learning environment

The virtual environment for training the agent is obtained pursuant to identifying an average virtual patient (AVP). The chosen model for AVP embraces a single delay model (SDM) of the glucose-insulin control system in Eq. (1). For the desire of the present paper, the AVP is representative of the type-2 diabetic patient, identified by the parameters shown in Table 1 [34,36,37]. The reason for using this large-scale model represented for the patients is that it has been accepted by the Food and Drug Administration (FDA) with the aim of not using animals for preclinical testing of artificial pancreas (AP) controlling [37]. Moreover, in the data of the patients 2 and 3, the τ_g is equal to zero to test the robustness of our model-free method.

Eqs. (3) and (4) represent the relationship between the constant parameters like V_I and G^*, and also patient-specific parameters represented in Table 1. The initial levels of glycemia, G_b (mM), and the insulinemia, I_b (pM), have been set as discussed in Refs. [37,38]. Consequently, the pair of model parameters T_{gh} and T_{iGmax} can be calculated according to Eqs. (3) and (4) [39].

$$T_{gh} = V_G G_b\left(K_{xg} + K_{xgi}I_b\right) \tag{3}$$

$$T_{igmax} = \frac{V_I K_{xi} I_b}{\varphi(G_b)} \tag{4}$$

The two introduced dynamics in the previous section are treated as follows. As it is realized, it takes time to absorb all amounts of the injected insulin. Due to the fact that the algorithm gets new action from the agent in each frame, it must be exactly calculated how much of the previous insulin injection has remained below the skin and has not been absorbed yet. Hence, according to the remaining insulin in the former steps and their absorption dynamics, the algorithm must learn how much insulin is needed after each step. This approach makes the insulin injection perform optimally and inhibits injecting excess insulin. Moreover, the aforementioned approach contributes to getting the simulation conditions closer to reality resulting in an optimal and robust controller. The simulator is also combined with a meal schedule to simulate a patient's behavior in a more realistic manner.

4.2. Formulation and reward function

Blood glucose metabolism is a dynamic system in which the glucose changes over time as the results of many factors such as food intake and insulin doses. The learning process of RL is based on the interaction between a decision-making agent and its environment, which will lead to an optimal action policy that results in desirable states. The block

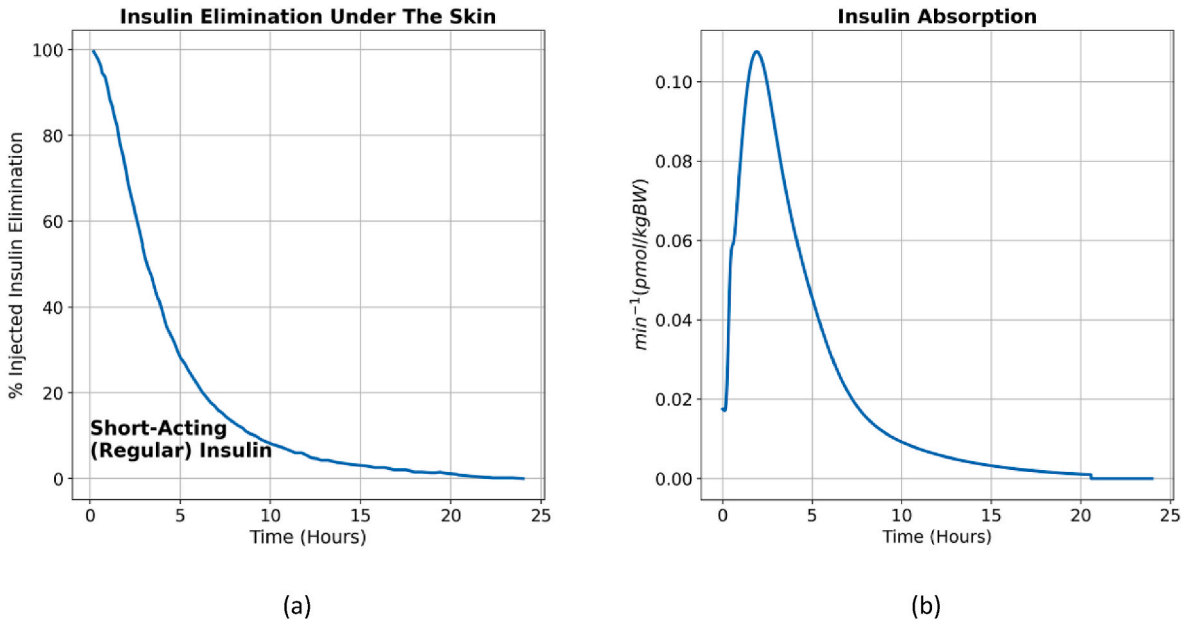

Fig. 1. (a) Insulin elimination and (b) absorption under the skin during a course of 24 h.

diagram of the entire process is demonstrated in Fig. 2.

Here, the problem of blood glucose control is framed as a Markov decision process (MDP) consisting of the 5-tuple (S; A; R; S'; done) states, including the blood glucose level along with the amount of remaining cumulative injected insulin in each time frame. The resolution at which data is collected from the simulator is 3 min. However, the agent observes the data every 120 min. The numbers are picked in a way so that it would represent the real situation as much as possible, making it a more viable solution compared to previous studies [40].

Actions are real positive numbers in the ranges described in Table 2 for each patient, denoting the volume of the insulin bolus to be injected in each time frame to control the blood glucose level in the range [4, 7.5] (mM).

Furthermore, the 'done' attribute is set to False by default except when the glucose level reaches either below 2.78 (mM) or above 13.9 (mM), indicating fatal states for the system. The environment is reset to new initial states each time the system reaches the fatal state during training.

Finally, the reward function at each time frame is designed as Eq. (5):

Table 2
Network hyperparameters used for training.

	Patient 1	Patient 2	Patient 3
layer size	80	80	100
batch size	512	512	768
Buffer size	10000	10000	10000
α	0.05	0.05	0.002
τ	0.1	0.1	0.02
γ	0.9	0.9	0.93
Update every	460	460	450
No of Updates	63	63	150
Batch normalization	False	True	False
No of Layers	2	5	2
Output Range (unit)	0–15	0–30	0–15

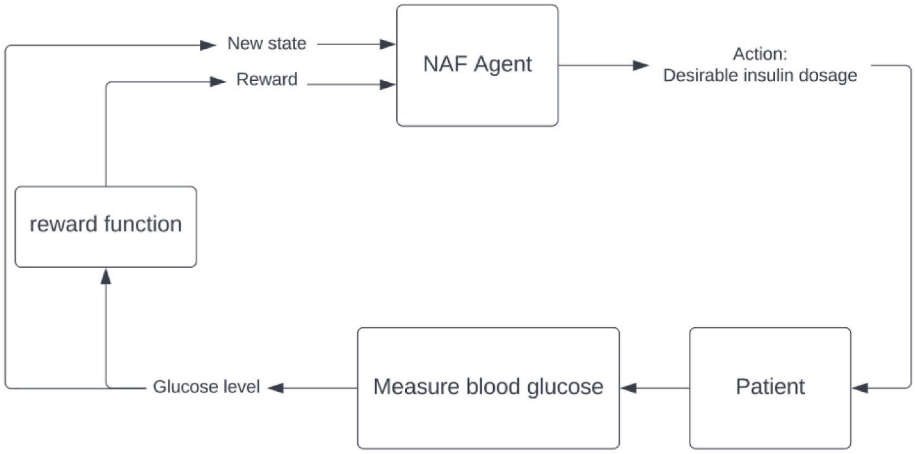

Fig. 2. Block diagram of the RL control process.

$$R(t) = \begin{cases} N(Gs)/24\ , 4 < Gs(t) < 7.5 \\ -1, Gs(t) < 2.78, Gs(t) > 13.9 \\ 0, otherwise \end{cases} \quad (5)$$

Where Gs is an array of glucose levels in each time frame, while $N(Gs)$ is the number of time frames in which the system was in the normoglycemic area in the past 24 h. We noticed outstanding performance using this function for the following reasons: 1. In fatal states, the agent receives a negative reward, which distinguishes them from normal states, making the agent avoid such states easily; 2. The performance of the whole day is taken into account, not just a single time frame.

Within the Q-learning framework [41], the state-action value function $Q(s,a)$ is learned through temporal-difference updates as Eq. (6):

$$Q(s_t, a_t) = R(s_t, a_t) + (1-\gamma) max_{a^*} Q(s_{t+1}, a^*) \quad (6)$$

From this Q function, one can extract the optimal policy as Eq. (7):

$$\pi^*(s_t) = argmax_a Q(s_t, a) \quad (7)$$

It is notable that this formulation requires discrete action bins, while the problem here requires continuous action space. For this matter, a method that can operate in continuous space, if trained properly, could outperform the ones with discrete actions [29] since the actions are less restricted. If we look at the problem from a regression perspective, deep neural networks are good candidates to accompany RL in order to tackle the issue.

To measure the utility of deep RL for the task of blood glucose control, we learned policies using NAF and tested these policies on simulated data across several different initial conditions and patients.

Different hyperparameters (HP) were used to maximize efficiency each time we trained our networks. The values of parameters used in each case can be found in Table 2 and were achieved through trial and error after thousands of hours of training and testing using Optuna, a library for automated HP tuning. We initialized our networks using PyTorch defaults and evaluated policies for 30 consecutive simulation days each time to maximize the confidence of the acquired results. The criterion used to assess the performance as well as HP tuning was the percentage of time spent in the gluconormal area in the most recent 24 h, but we also took into account how fast the agent can bring the glucose level in this area. The results are reported in Section 5.

4.3. NAF agent

The purpose of NAF is to derive a continuous variant of the Q-learning algorithm as an alternative to policy gradient and actor-critic methods, which are the typical choices when it comes to continuous control tasks. Using such task-specific representations dramatically improves efficiency but limits the range of tasks that can be learned and requires more expert domain knowledge. On the other hand, using model-based RL also improves efficiency, but limits the policy to only be as good as the learned model. It is therefore desirable to bring the generality of model-free deep reinforcement learning into real-world domains by reducing their sample complexity.

While a number of representations are possible that allow for analytic maximization, the one used in this study is based on a neural network that separately outputs a value function term $V(x)$ and an advantage term $A(x,u)$, which is parameterized as Eq. (8) a quadratic function of nonlinear features of the state. The summation of $A(x,u)$ and $V(x)$ results in $Q(x,u)$ demonstrated as Eq. (9)

$$A(x,u) = -1/2[u-\mu(\theta^\mu)]^T P(x|\theta^P)[u-\mu(\theta^\mu)] \quad (8)$$

$$Q(x,u) = A(x,u) + V(x) \quad (9)$$

Where

x: Observations
u: Action
Q: Q-Learning function
A: Actor function
V: Value function (neural network)
μ: Predicted actions based on the actor (neural network)
L: Lower-triangular matrix (neural network)

This representation is more restrictive than a general neural network function since the Q-function is quadratic in u. The action that maximizes the Q-function is always given by $\mu(x|\theta)$. NAF also avoids the need for a second actor or policy function, resulting in a simpler algorithm. The simpler optimization objective and the choice of value function parameterization result in a substantially more sample-efficient algorithm when used with large neural network function approximations on a range of continuous control domains. Hence, NAF is much easier to deal with, compared to DDPG. Furthermore, NAF is the first method to combine normalized action-value functions [42] and decompose Q into a state-value term V and an advantage term A [43,44], along with deep neural networks into an algorithm that can be used to learn policies for a range of challenging continuous control tasks [30].

4.4. Implementation

The entire implementation is done in Python language. The PyTorch library is used to train and test neural networks, and Optuna is used for HP optimization. Algorithm 1 shows the main procedure of implementation in detail. A flowchart also demonstrates the procedure in Fig. 3.

Algorithm 1. NAF training for blood glucose control

5. Setup and evaluation

NAF agents are trained in the environment explained in Section 4.1 and with random initial conditions for each episode. The RL models are trained using ε-greedy exploration with $\varepsilon = 0.9$. The mean square error (MSE) loss of our temporal difference predictions was also optimized using the Adam optimizer. Each model was trained for a maximum 20000 frames. After every 2000 frames, the network was evaluated; and in case of getting satisfactory results, the training was stopped. While the duration of each test was 30 days, we evaluated the performance of the models based on the percentage of time the simulated patients' glucose level was kept in the desired range during both the first and the final 24 h, as discussed before.

To evaluate our proposed method, three different experiments are conducted, each with different sets of environment parameters that equally mean three different virtual patients. The network and environment parameters used for training in each experiment are reported in Tables 1 and 2

In order to simulate a trial, a periodic daily meal protocol was designed consisting of 3 meal intakes each day at 9:00 a.m., 1:00 p.m., and 9:00 p.m. with the glucose level set to 9, 10, and 10.5 (mM), respectively, for each intake. Throughout the experiment, the performance of our method is tested with the meal intake, and then each virtual patient was tested to assess the performance of the algorithm in a variety of environments characterized by different parameters.

The first experiment included a delay of 6.5 min for insulin infusion (represented by the parameter τ_g). The initial values for glucose and insulin in the evaluation process were set to 8.85 mM and 58.95 pM, respectively.

In the next experiment, as seen in Table 1, some of the environment parameters were changed, resulting in different behavior. For instance, the delay was negligible and so was set to zero, as mentioned in Section 4.1. Also, the overall insulin level, including the initial value, was much lower (almost half of that of the first experiment). This entailed some

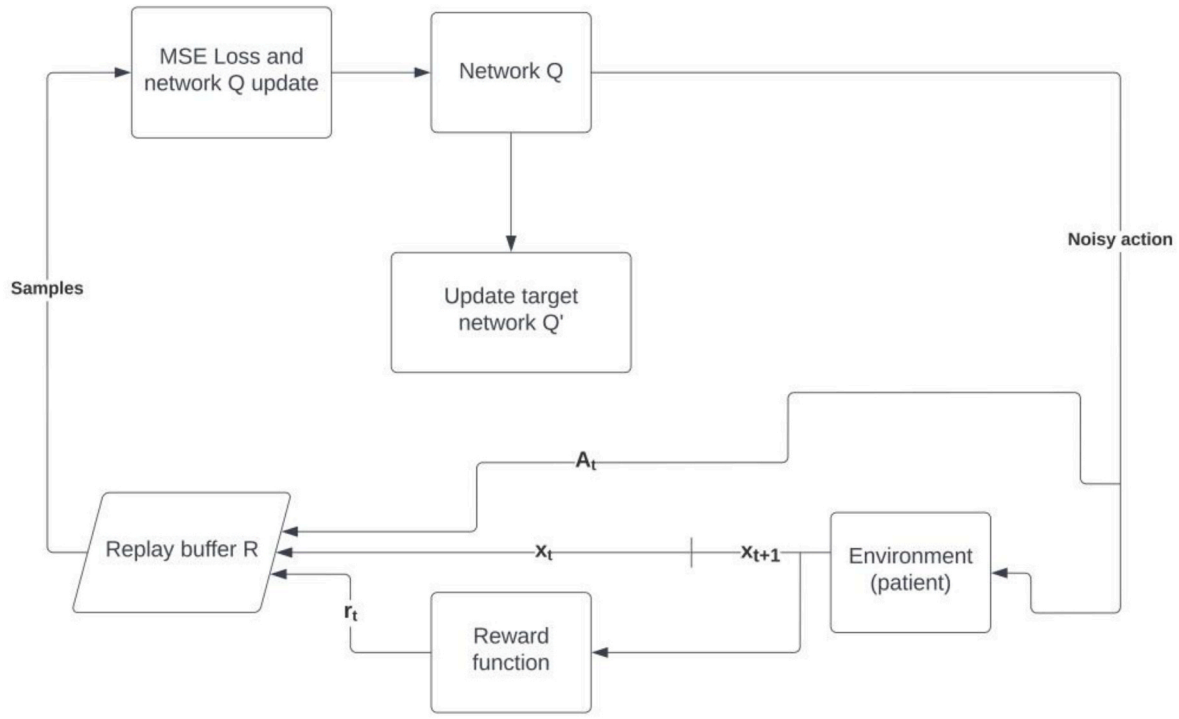

Fig. 3. Implementation flowchart.

Randomly initialize normalized Q network
Randomly initialize environment E
Initialize target network Q' with weight $\theta^{Q'} \leftarrow\leftarrow \theta^{Q}$
Initialize empty replay buffer R
Receive initial observation state x_1
For episode: $1, M$ do
  For every time frame: $1, t$ do
    Get noisy action u_t from network Q
    Execute $E(u_t)$ and observe r_t, x_{t+1}
    Store transition (x_t, u_t, r_t, x_{t+1}) in R
    For iteration $1, i$ do
      Sample a random mini-batch of m transitions from R
      Set $y_i = r_i + \gamma V'(x_{i+1} | \theta^{Q'})$
      Update θ^{Q} by minimizing the loss: MSE$(y_i, Q(x_{i+1}, \theta^{Q'}))$
      Update the target network: $\theta^{Q'} \leftarrow\leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$
    End
  End
End

changes in our network HP for training. More specifically, the number of hidden layers was increased and batch normalization was enabled to achieve a good performance.

Finally, the third set represented another case of a diabetic patient. Like the previous experiment, the delay was set to zero. Also, the maximal rate of second-phase insulin release (T_{iGmax}) and the first-order rate of insulin disappearance (K_{xi}) were substantially higher compared to the second case. This results in a more challenging control task. To handle this complexity, the network architecture was changed, as the output range was broadened so that the algorithm has more choices for insulin injection. Also, the number of layers, the learning rate, and τ were decreased.

6. Results and discussion

The glucose levels of all virtual patients are shown in Fig. 4 for comparison. As can be seen, the algorithm was able to achieve the desired behavior in all the three patients, keeping the glucose level in the range of normoglycemia, apart from the times of meal intake, which is sensible. It is also noteworthy that the injected insulin successfully brought down the glucose level after each meal with noticeable speed, entering the range in 3–4 h. All of these effects were achieved without

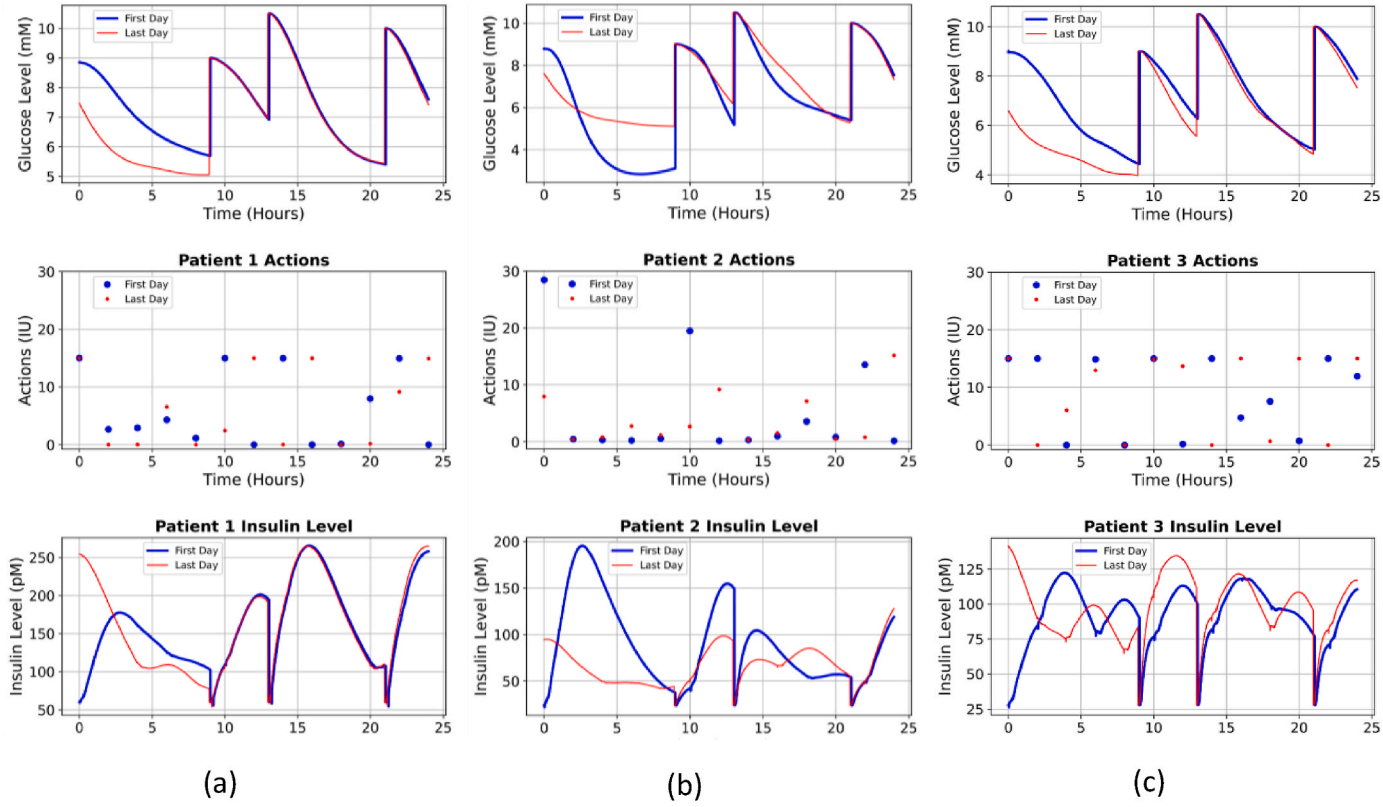

Fig. 4. State-action diagrams for AVPs in result of glucose control with NAF.

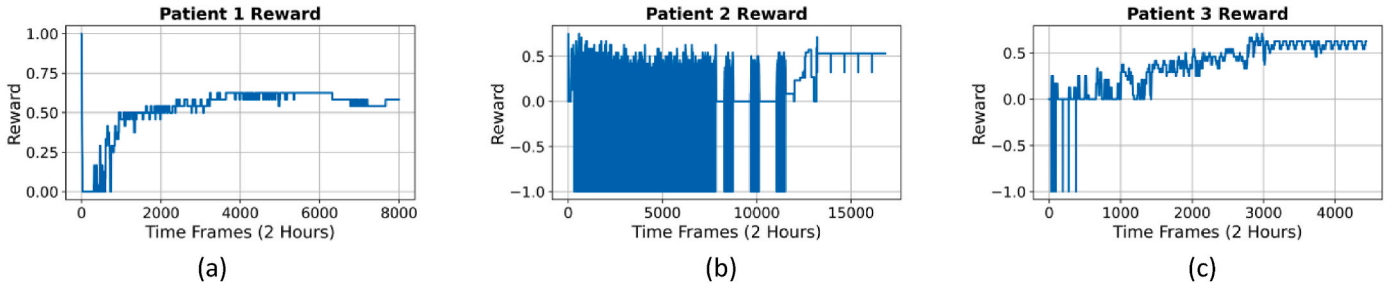

Fig. 5. Rewards gained during training the NAF agent in each frame for each virtual patient.

any dangerous fluctuations in glucose figures or high doses of insulin infusion, causing no hypoglycemic or hyperglycemic events. Finally, the reward values for experiments are shown in Fig. 5. It is evident that through exploration the agent was able to avoid fatal states and optimize the actions. One major advantage of our proposed method is its ability to achieve robust performance without meal announcements and with a sampling resolution of 2 h, making it a suitable candidate to be tested in real-life situations. To be more specific, insulin and glucose levels are fed to the agent, and actions are received every 2 h.

Turning to actions taken by the agent, some interesting insights can be found. Firstly, for patient 1 represented in Fig. 4 (a) the agent understood that there is a delay between taking actions and insulin absorption; therefore, some actions were taken before the glucose surges, caused by meal intakes, to take the glucose level down later on. Secondly, as a result of using our reward function, the agent decided that it is not always necessary to inject high doses of insulin after meal intakes but instead take the fluctuations of the past 24 h and previous insulin injections into account to take the situation under control. This approach also results in the efficient injection of insulin. It is a common practice in the medical community to inject relatively high doses after each meal. Our study shows that this is neither necessary nor efficient to always take such actions. The states of the two other patients are illustrated in Fig. 4 (b), 4 (c). It can be seen that although having different model parameters for the patients, the agent could successfully identify their behavior and calculate the necessary doses of insulin to inject to regulate the glucose level. The outstanding performance of the agent can be clearly seen in Fig. 4 (c). Patient 3 has a higher disappearance rate constant for insulin (K_{xi}) in comparison to the others. Therefore, the agent decided to inject higher insulin doses to regulate the glucose level. The glucose level curves of all three patients are almost identical for the last day of the simulation, but the action profiles are still very different. This verifies the intelligence and the learning ability of our proposed controller. For the first day of the simulation, it is also evident that the agents could successfully take the patients' glucose level from different initial conditions to the desirable range by injecting proper insulin doses. Actually, the difference between the action values of the first and the last days comes from these different initial conditions. It is noteworthy that the controller could make all the patients follow the same glucose level profile in the last day, although having different dynamic model for each one.

In order to compare the performance, a sliding mode controller (SMC) is selected as a typical nonlinear conventional controller and is applied to the problem. Firstly, conventional controllers are model-based, meaning they require a mathematical model of the system. However, in the case of our problem, it is obvious that there is no model available for every patient in reality. In clinics, the rate at which glucose feedback values can become available for the controller is about every 2 h, the same as the insulin injection rate. Thus, for more realistic simulation and comparison purposes, the time step of the SMC is set to 2 h. Note that such controllers need a value to regulate the system. This value is chosen as the average of the desirable range in Eq. (5) i.e. $G_d = 5.75$. Fig. 6 shows the result of applying SMC to the glucose-insulin system. It can be seen when the glucose level is high and far from the desired value, the controller injects relatively high doses of insulin. While the glucose level becomes close to the desired value, the injected insulin becomes lower. If the glucose level reaches the desired level (or becomes lower), the controller does not inject any insulin; but, because of the delayed effect of insulin absorption under the skin, the glucose level continues to decrease further and even crosses the desirable range. The SMC could not infer and handle the effect of the delay, learn from the past actions and predict the future in contrast to the NAF controller. SMC only uses the instantaneous feedback values for decision-making; thus, the injection pattern and the action values are not efficient in comparison to the output of the NAF controller. Also, the robustness of SMC could not handle the parameter difference between the patients since their differences are not slight. Consequently, SMC cannot be employed with the same parameters for all patients, making it useless in practice.

Beyond the performance of the learned policies, across our experiments, we found that over five hundred days of simulation data were required when training our deep approaches. While this is infeasible in real-life situations outside of simulation, since our approach does not rely on meal announcement or expert knowledge to work properly, we argue that an accurate simulation can be sufficient, and the learning phase does not need to be done with real data. This, of course, entails using an accurate mathematical model with parameters tuned for each patient separately. Furthermore, lack of real data could be an issue when applying this approach to treatment of other diseases. Such cases require real data, either from a dataset or real-time data, which naturally seems to be very difficult and time-consuming to collect. It is also worth pointing out that the data provided and the simulated environment's accuracy delineate the trained model's performance. This means that any disturbance or inaccuracy that was not accounted for in the modeling of the dynamics is also absent in the training process. Therefore, an extensive evaluation is indeed necessary before putting such algorithms to practice. Some ongoing works seem to take extra precautions and set strict constraints to make the decision-making process as safe as possible [45]. However, health applications are often safety-critical. Thus, the susceptibility of deep RL to unexpected or unsafe behavior can pose a significant risk [46]. The technique used here to make the study more tractable was to modify the reward function, using a negative termination penalty to discourage dangerous behavior as much as possible. Nevertheless, it is important that practitioners take every step to evaluate the safety of their approaches.

7. Conclusion and future works

In this study, a rather new RL algorithm was put to use to control the blood glucose level of type-2 diabetes patients. An application based on NAF has been put to the test for daily insulin administration. The purpose of NAF was to suggest levels of insulin to be injected at each time frame in order to minimize the cost induced by hypo- and hyperglycemic events. NAF has proved a promising control approach, able to perform efficiently in the presence of meal estimation and improves glycemic control compared to standard optimized open-loop basal-bolus therapies. The algorithm has been designed and developed in a model-free approach in order to avoid additional inaccuracies and parameter uncertainty introduced by the mathematical models of the glucoregulatory system. Also, the simulation environment was set up to accurately imitate the basal-bolus process while being applicable in real-time clinical situations. This, alongside our network model and reward function, resulted in an intelligent and robust system that successfully performed the controlling task across different patients in our test. Finally, unlike similar studies, a more accurate model containing infusion dynamics was used in training by combining the insulin absorption model with the model described by Eqs. (1) and (2). This, along with the fact that the administrated insulin dose is in *IU* which is the most

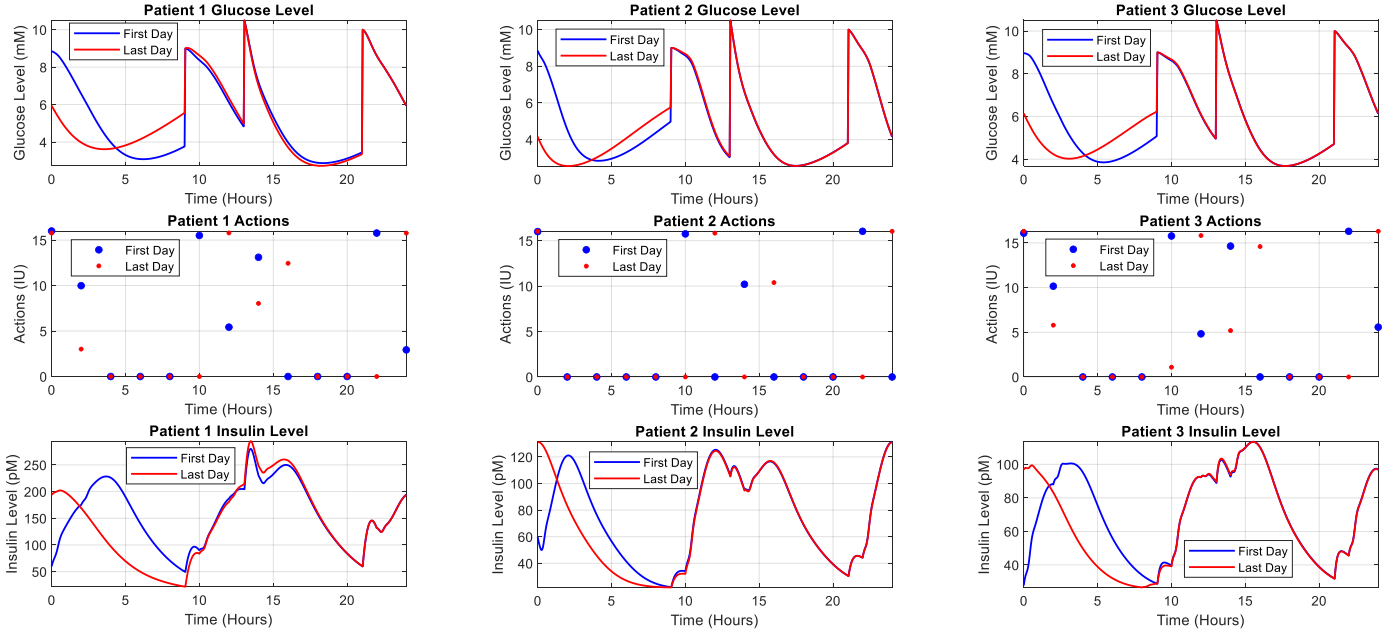

Fig. 6. State-action diagrams for AVPs in result of glucose control with SMC.

common unit used in the clinical literature, makes this approach more applicable for practitioners.

In our prospects, two horizons are yet to be explored. Firstly, an extensive investigation of new training settings can be considered. This could include finding better-tuned hyperparameters for faster convergence as well as improving the reward function even further in order to achieve an optimal solution guaranteed to minimize drug dosage while regulating a patient's glucose level as quickly as possible. Secondly, the proposed approach can be applied to similar applications in healthcare formulated as a problem of controlling a medical condition through exposing the patient to sequential treatment such as drug delivery or radiotherapy. Some notable candidates could be the treatment of Sepsis through intravenous administration and the treatment of chronic diseases such as epilepsy using observed electroencephalograph signals as states. Lastly, a field experiment can be conducted under strict supervision in order to test the limitations and merits of this approach outside of simulated environments. This, for instance, can be done by creating a mobile or web application that accepts glucose levels and patient parameters and suggests a regular dosage of insulin to be injected.

Data availability

The source code/datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Acknowledgment

The authors are grateful to Dr. Ali Zamani, Dr. Reyhaneh Naseri, and Dr. Amir Mohammad Farrokhi, for their advice during this study.

APPENDIX 1

The definitions of the parameters used in Eq. (1) are as follows [39]:

- $K_{xg}, [min^{-1}]$, is the rate of insulin-independent glucose uptake.
- $K_{xgi}, [min^{-1}pM^{-1}]$, is the rate of glucose uptake by tissues (insulin-dependent) per pM of plasma insulin concentration.
- $T_{gh}, [min^{-1}(mmol/kgBW)]$, is the net balance between hepatic glucose output and insulin-independent zero-order glucose tissue uptake (mainly by the brain, supposed constant throughout the experiment).
- $V_G, [L/kgBW]$, is the apparent distribution volume for glucose.
- $K_{xi}[min^{-1}]$, is the apparent first-order disappearance rate constant for insulin.
- $T_{iGmax}, [min^{-1}(pmol/kgBW)]$, is the maximal rate of second-phase insulin release.
- $V_I, [L/kgBW]$, is the apparent distribution volume for insulin.
- $\gamma, [\sharp]$, is the progressivity with which the pancreas reacts to circulating glucose concentrations. If γ were zero, the pancreas would not react to circulating glucose at all; if γ were 1, the pancreas would respond according to a Michaelis-Menten dynamics, where G^* is the glucose concentration of half-maximal insulin secretion; when γ is greater than 1 (as is usually the case), the pancreas responds according to a sigmoidal function.
- $G^*, [mM]$, is the glycemia at which the insulin release is the half of its maximal rate; at a glycemia equal to G^* corresponds an insulin secretion equal to $T_{iGmax}/2$;

APPENDIX 2

Fig. A-1 (a) demonstrates the Insulin elimination in IU [35]. This curve is inverted and multiplied by 6.944 to obtain the insulin absorption in $pmol$ as in Fig. A-1 (b). The time derivative of Fig. A-1 (b) yields insulin absorption in $pmol/h$. Another unit conversion must be done according to Eq. (A-1) to construct the insulin input for Eq. (1), which is $(pmol/kgBW)min^{-1}$. Fig. A-1 (c) represents the final insulin input dynamics.

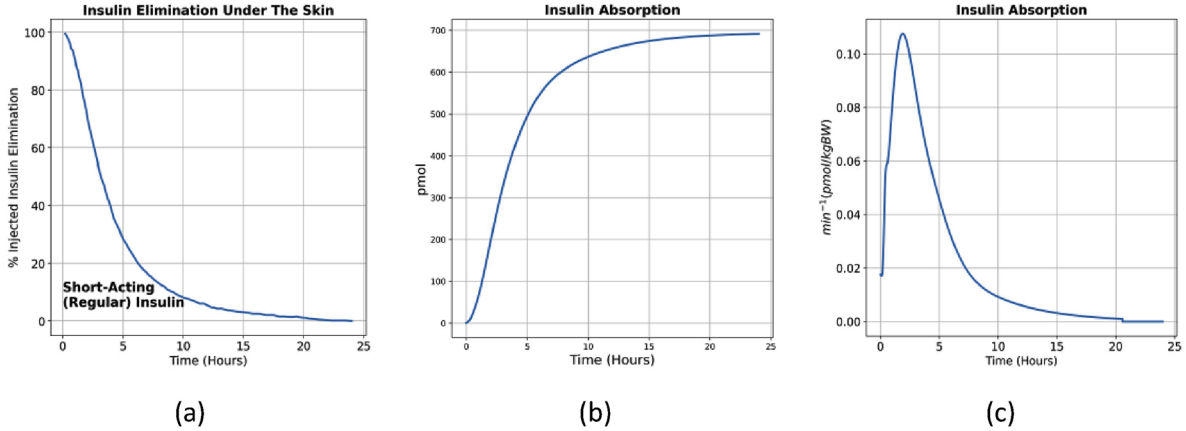

Fig A-1. Steps for obtaining insulin absorption dynamics.

$$(pmol/h) * V_i(L/kgBW) * \frac{1h}{60min} * \frac{1}{5.5L(blood)} = (pmol/kgBW)min^{-1} \quad \text{(A-1)}$$

References

[1] C.M. Wiener, et al., Harrison's Principles of Internal Medicine : Self-Assessment and Board Review, 2018.
[2] M. Shifrin, H. Siegelmann, Near-optimal insulin treatment for diabetes patients: a machine learning approach, Artif. Intell. Med. 107 (2020), 101917.
[3] P. Palumbo, et al., Time-delay model-based control of the glucose–insulin system, by means of a state observer, Eur. J. Control 18 (6) (2012) 591–606.
[4] Z. Wu, et al., An effective approach for the protection of user commodity viewing privacy in e-commerce website, Knowl. Base Syst. 220 (2021), 106952.
[5] Z. Wu, et al., Constructing dummy query sequences to protect location privacy and query privacy in location-based services, World Wide Web 24 (1) (2021) 25–49.
[6] W. Yan, et al., Extracting diverse-shapelets for early classification on time series, World Wide Web 23 (6) (2020) 3055–3081.
[7] F.S. Gharehchopogh, P. Mohammadi, P. Hakimi, Application of decision tree algorithm for data mining in healthcare operations: a case study, Int. J. Comput. Appl. 52 (6) (2012).
[8] F.S. Gharehchopogh, P. Mohammadi, A case study of Parkinson's disease diagnosis using artificial neural networks, Int. J. Comput. Appl. 73 (19) (2013).
[9] F.S. Gharehchopogh, M. Molany, F.D. Mokri, Using artificial neural network in diagnosis of thyroid disease: a case study, International Journal on Computational Sciences & Applications (IJCSA) 3 (2013) 49–61.
[10] S. Wang, et al., Multi-scale context-guided deep network for automated lesion segmentation with endoscopy images of gastrointestinal tract, IEEE Journal of Biomedical and Health Informatics 25 (2) (2020) 514–525.
[11] J.J. Khanam, S.Y. Foo, A comparison of machine learning algorithms for diabetes prediction, ICT Express 7 (4) (2021) 432–439.
[12] G.M. Steil, Algorithms for a closed-loop artificial pancreas: the case for proportional-integral-derivative control, Journal of diabetes science and technology 7 (6) (2013) 1621–1631.
[13] B.W. Bequette, A critical assessment of algorithms and challenges in the development of a closed-loop artificial pancreas, Diabetes Technol. Therapeut. 7 (1) (2005) 28–47.
[14] E. Atlas, et al., MD-logic artificial pancreas system: a pilot study in adults with type 1 diabetes, Diabetes Care 33 (5) (2010) 1072–1076.
[15] I. Fox, J. Wiens, Reinforcement Learning for Blood Glucose Control: Challenges and Opportunities, 2019.
[16] S.K. Garg, et al., Glucose outcomes with the in-home use of a hybrid closed-loop insulin delivery system in adolescents and adults with type 1 diabetes, Diabetes Technol. Therapeut. 19 (3) (2017) 155–163.
[17] J.L. Ruiz, et al., Effect of insulin feedback on closed-loop glucose control: a crossover study, Journal of diabetes science and technology 6 (5) (2012) 1123–1130.
[18] D. Silver, et al., A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, Science 362 (6419) (2018) 1140–1144.
[19] A. Rajeswaran, et al., Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations, 2017 arXiv preprint arXiv: 1709.10087.
[20] M. Tejedor, A.Z. Woldaregay, F. Godtliebsen, Reinforcement learning application in diabetes blood glucose control: a systematic review, Artif. Intell. Med. 104 (2020), 101836.
[21] J. Schulman, et al., Trust region policy optimization, in: International Conference on Machine Learning, PMLR, 2015.
[22] M. Komorowski, et al., The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care, Nat. Med. 24 (11) (2018) 1716–1720.
[23] W.-H. Weng, et al., Representation and Reinforcement Learning for Personalized Glycemic Control in Septic Patients, 2017 arXiv preprint arXiv:1712.00654.
[24] N. Prasad, et al., A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units, 2017 arXiv preprint arXiv:1704.06300.
[25] P. Klasnja, et al., Efficacy of contextually tailored suggestions for physical activity: a micro-randomized optimization trial of HeartSteps, Ann. Behav. Med. 53 (6) (2019) 573–582.
[26] I. Clavera, et al., Learning to Adapt: Meta-Learning for Model-Based Control, 2018, p. 3, arXiv preprint arXiv:1803.11347.
[27] R. Visentin, et al., The university of Virginia/Padova type 1 diabetes simulator matches the glucose traces of a clinical trial, Diabetes Technol. Therapeut. 16 (7) (2014) 428–434.
[28] P.S. Thomas, E. Brunskill, Importance sampling with unequal support, in: Thirty-first AAAI Conference on Artificial Intelligence, 2017.
[29] P.D. Ngo, et al., Control of blood glucose for type-1 diabetes by using reinforcement learning with feedforward algorithm, Comput. Math. Methods Med. 2018 (2018), 4091497.
[30] S. Gu, et al., Continuous deep q-learning with model-based acceleration, in: International Conference on Machine Learning, PMLR, 2016.
[31] T. Haarnoja, et al., Composable deep reinforcement learning for robotic manipulation, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018.
[32] S. Gu, et al., Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017.
[33] C. Cobelli, et al., An integrated mathematical model of the dynamics of blood glucose and its hormonal control, Math. Biosci. 58 (1) (1982) 27–60.
[34] M. Di Ferdinando, et al., Sampled-data static output feedback control of the glucose-insulin system, IFAC-PapersOnLine 53 (2) (2020) 3626–3631.
[35] E.D. Lehmann, et al., Incorporating a generic model of subcutaneous insulin absorption into the AIDA v4 diabetes simulator: 1. A prospective collaborative development plan, Journal of diabetes science and technology 1 (3) (2007) 423–435.
[36] A. Borri, et al., Sampled-data observer-based glucose control for the artificial pancreas, Acta Polytechnica Hungarica 14 (1) (2017) 79–94.
[37] P. Palumbo, et al., Model-based control of plasma glycemia: tests on populations of virtual patients, Math. Biosci. 257 (2014) 2–10.
[38] C. Dalla Man, R.A. Rizza, C. Cobelli, Meal simulation model of the glucose-insulin system, IEEE Trans. Biomed. Eng. 54 (10) (2007) 1740–1749.
[39] P. Palumbo, S. Panunzi, A. De Gaetano, Qualitative behavior of a family of delay-differential models of the glucose-insulin system, Discrete & Continuous Dynamical Systems-B 7 (2) (2007) 399.
[40] I. Fox, et al., Deep reinforcement learning for closed-loop blood glucose control, in: Machine Learning for Healthcare Conference, PMLR, 2020.
[41] C.J. Watkins, P. Dayan, *Q-learning.* Machine learning 8 (3–4) (1992) 279–292.
[42] K. Rawlik, M. Toussaint, S. Vijayakumar, On stochastic optimal control and reinforcement learning by approximate inference, in: Twenty-third International Joint Conference on Artificial Intelligence, 2013.
[43] M.E. Harmon, L. Baird, A.H. Klopf, Advantage updating applied to a differential game, Adv. Neural Inf. Process. Syst. (1995) 353–360.
[44] M.E. Harmon, L.C. Baird III, Multi-player Residual Advantage Learning with General Function Approximation, Wright Laboratory, WL/AACF, Wright-Patterson Air Force Base, OH, 1996, p. 45433, 47308.
[45] J. Futoma, M.A. Masood, F. Doshi-Velez, Identifying distinct, effective treatments for acute hypotension with SODA-RL: safely optimized diverse accurate reinforcement learning, AMIA Summits on Translational Science Proceedings (2020) 181, 2020.
[46] J. Leike, et al., AI Safety Gridworlds, 2017 arXiv preprint arXiv:1711.09883.