2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Deep Deterministic Policy Gradient Control of Type 1 Diabetes

Federico BALDISSERI*, Danilo MENEGATTI, and Andrea WRONA

***Abstract*— Type 1 diabetes is one of the major concerns in current medical studies. Traditional clinical practice involves non-autonomous manual injection of insulin in the blood, while current research in the field of autonomous regulation of blood glucose concentration mostly focuses on model-based control techniques. This paper introduces a novel Reinforcement Learning-based controller for autonomous glycemic regulation in the treatment of type 1 diabetes, building on the Deep Deterministic Policy Gradient algorithm. The proposed control method is validated through in-vitro simulations on the Bergman glucoregulatory model, proving that it successfully preserves healthy values of blood glucose concentration, while overcoming both standard clinical practice and classical model-based control techniques in terms of both control effort and computational efficiency for real-time applications.**

***Index Terms*— Deep Reinforcement Learning, Deep Deterministic Policy Gradient, Type 1 Diabetes.**

I. INTRODUCTION

A. Motivations

Diabetes is one of the most widespread and impactful diseases in our society. The World Health Organization estimates that over 420 million people worldwide (i.e., about 9.1% of the adult population) have diabetes, and that this number will more than double by 2030. Moreover, diabetes is a major cause of death: an estimated 6.7 million people died from such disease in 2021. Lastly, almost 376 billion dollars per year are spent by countries worldwide for medical care and drugs provisioning [1].

Thus, this work is motivated by the illustrated severity of clinical conditions of diabetes, the current incidence of such disease and its estimated rise, and its onerous repercussions on individuals, families, healthcare systems, and countries.

In particular, this work focuses on type 1 diabetes, that is characterized by the autoimmune destruction of beta cells in the pancreas, that is responsible for producing insulin, namely the hormone essential for facilitating absorption of glucose from the bloodstream into cells throughout the body. Therefore, the patient is completely dependent on external insulin administration. A safe and effective glucose regulation, maintaining blood glucose concentration in the healthy range $70 - 140$ mg/dl, is of paramount importance since hyperglycemia (i.e., high blood concentration) can lead to cardiovascular diseases and blindness, while hypoglycemia (i.e., low blood glucose concentration) can lead to coma and death [2].

All authors are with the Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy. Emails: {baldisseri, menegatti, wrona}@diag.uniroma1.it

*Corresponding author. Email: baldisseri@diag.uniroma1.it

This work has been partially carried out in the framework of the CADUCEO project (No. F/180025/01-05/X43), supported by the Italian Ministry of Enterprises and Made in Italy.

B. Related works

Diabetes is traditionally self-managed by the patient, through glucose concentration measurements and manual administration of insulin via injection or pumps, entailing inconvenient complications in daily life. An attempt to overcome this issue was performed with the development of the Artificial Pancreas [3], that consists of a continuous glucose monitoring (CGM) system, an insulin pump, and a control algorithm that produces instructions for the pump based on the CGM inputs. Figure 1 shows the Artificial Pancreas control framework.

Several classes of control algorithms have been used in academic literature. Proportional integral derivative (PID) control is a classical logic that provides simple real-time control adjustments, yet is not best suited to complex nonlinear dynamics [4]. In [5], authors exploit a Neural Network for tuning the gains of the PID controller. Fuzzy-logic based control shows good flexibility for patient personalization, but has limited predictive capabilities [6]. While linear parameter-varying control can accommodate over time to the patient's changing physiological conditions, it may also necessitate the use of computational resources that devices with low processing power may not have [7]. In [8], a Kalman Filter approach is used to estimate blood glucose levels and adjust insulin dosages accordingly; Kalman Filters can handle measurement noise and uncertainties in the system dynamics, making them robust to inaccuracies in sensor readings and modeling errors; nevertheless, they are sensitive to initial conditions and parameter values, thus small errors in the initial state estimate or covariance matrix can propagate over time and affect performances. Model Predictive Control (MPC) has been extensively enforced, due to its capability to handle receiding time horizon control as a constrained optimization problem [9]–[11]; nevertheless, models identified

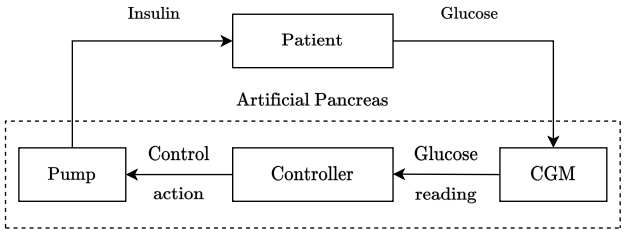

Fig. 1. Block scheme of the Artificial Pancreas control framework.

through MPC are often difficult to implement in real-time because of their complexity and need of an online model-based optimization [12]. In [13], authors integrate MPC with a Neural Network in order to better address inter/intra patient variability.

In recent years, Reinforcement Learning (RL) control has been broadly investigated due to its potential to develop fully automated algorithms that can provide personalized control under varying conditions [14]. Moreover, Deep Reinforcement Learning can be enforced in order to leverage deep neural networks to automatically learn meaningful representations from high-dimensional and complex data, thus being more efficient in learning optimal control policies. As an example, in [15], [16] it is shown how it is possible to implement a Deep Q-Learning control strategy with a discretization of the insulin infusion, thus building an autonomous agent selecting dynamically the best insulin level to be injected in blood during the day.

However, none of the previously mentioned works has proposed a data-driven insulin control strategy leveraging continuous action spaces, thus avoiding a quantization which usually yields information losses [17].

This work overcomes the highlighted limitation in previous academic literature: its main contributions rely on (i) the application of a continuous control logic based on Deep Reinforcement Learning [18] to the problem of autonomous insulin regulation for patients suffering from type 1 diabetes, and (ii) the comparison of the proposed method with standard clinical practice and traditional model-based control logics such as Proportional Integral Derivative control and Nonlinear Model Predictive Control.

C. Document organisation

The remainder of the paper is organized as follows. Section II introduces the proposed control methodology; Section III describes the Bergman dynamical model of a patient suffering from type 1 diabetes; Section IV presents and discusses the simulations performed in order to validate the proposed method; Section V draws conclusions and outlines future research directions.

II. Control Methodology

A. Reinforcement Learning

Reinforcement Learning (RL) is a subclass of Machine Learning that focuses on training intelligent agents to enforce actions over an environment in order to maximize a cumulative reward, while monitoring state observations and immediate rewards [19]. Figure 2 shows the RL control framework, which is a closed-loop one, with the controller (or agent) computing the control signal based on two feedback signals: the observable state and the reward.

A RL problem can be formalized through a Markov Decision Process (MDP), that is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ where:

- $\mathcal{S}$ is the state space;
- $\mathcal{A}$ is the action space;
- $P = P(s'|s,a)$ is the state transition function, namely the probability that the environment transitions from state s to state s' when the agent chooses action a;
- $R(s,a,s')$ is the immediate reward that the agent gets when transitioning from state s to s' after taking action a;
- $\gamma \in [0,1)$ is a discount factor, representing the agent's preference for immediate rewards ($\gamma \approx 0$) over the future ones ($\gamma \approx 1$) .

The objective in a MDP is to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that specifies which action must be taken in each state, in order to maximize the state-value function defined as follows:

$$V_\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s]. \quad (1)$$

The latter quantity as a matter of fact represents the expected discounted cumulative reward the agent gets following a certain policy, considering a possibly infinite horizon.

MDPs can be handled through a variety of algorithms belonging to the class of dynamic programming, but only when the transition probability function is fully modeled and, hence, known. When $P(s'|s,a)$ is not known, meaning that the agent is not aware of the differential stochastic equations governing the environment's dynamics, MDPs must be tackled through RL [19]. Due to this property, in RL exploration is essential for discovering unknown states, transitions, and the associate rewards, thereby improving the agent's understanding of the environment dynamics. Exploration, which is carried out through the application of random actions over the environment, allows the agent to gather data necessary for accurate estimation of state-action values. Furthermore, exploration is crucial for discovering rare or hidden states that might lead to high rewards. Without sufficient exploration, the agent may prematurely converge to sub-optimal policies, resulting in stagnation or poor performance. The exploration must be balanced with the so–called exploitation, which involves leveraging the current knowledge to maximize short-term rewards. Exploitation is crucial for exploiting known high-reward strategies and achieving efficient performance once the agent has acquired sufficient knowledge about the environment. While exploitation aims to maximize immediate rewards, premature exploitation can hinder the agent's ability to discover better strategies. As it happens with undue exploration, over-reliance on exploitation can lead to sub-optimal policies, especially in environments with time-varying stochastic dynamics.

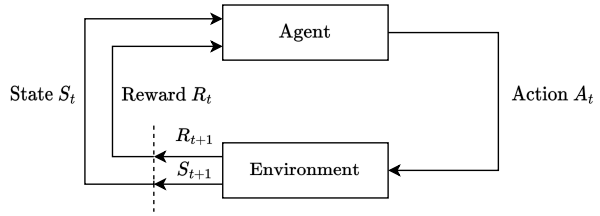

Fig. 2. Block scheme of the Reinforcement Learning control framework.

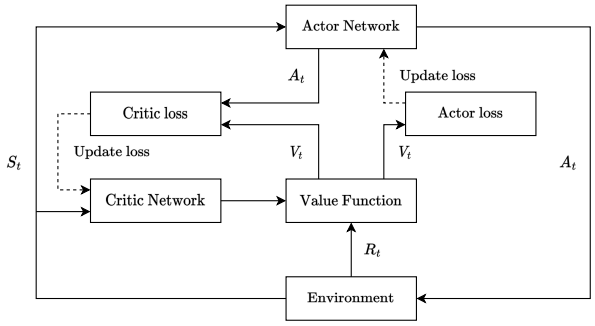

Fig. 3. Block scheme of the Deep Deterministic Policy Gradient control framework.

In related literature there are numerous algorithms to solve a RL problem, learning optimal policies. However, most of them are characterized by discrete states and actions, making them unfeasible to tackle traditional control problem, in which the control input and observable states continuously change over time.

B. Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) is an advanced Deep Reinforcement Learning algorithm that is able to cope with continuous state and action spaces. It is an extension of the popular Deterministic Policy Gradient algorithm [20], combining deep neural networks with the principles of actor-critic methods. It uses neural networks as approximation functions to learn the optimal policy (actor network), while continuously evaluating the action quality during the training phase (critic network). In particular, the agent is denoted by four neural networks: the first two are the actor and critic networks, whereas the last two are the so-called target actor and target critic, copies of the first two with slowly updated parameters used to stabilize the learning phase. To further improve sample efficiency and break the temporal correlations in the data, DDPG employs an experience replay buffer, in which past experiences (state, action, reward, next state) are stored. Random samples from this buffer constitute the training set for the neural networks.

As for the exploration strategy during training, this is usually carried out adding a random noise to the output of the actor network, thus applying an action which is slightly modified with respect to current optimal one. To achieve a proper balance between exploration and exploitation, the amplitude of the noise decreases over training episodes, thus focusing mostly on exploration of the environment at the start, and mostly on exploitation of the learned strategy at the end.

Figure 3 shows the DDPG control framework. The DDPG methodology is outlined in Algorithm 1.

Note that the agent is trained over a certain number of episodes E, each one lasting T steps. Over the internal steps, the agent selects an action and observes the state and the reward, then it stores the transition into the replay buffer. The learning step is performed both for the critic and the actor network by sampling random minibatches from the replay buffer. The last operation consists of updating the target networks through the parameter τ, which specifies to what extent the target networks should trust the current weights of the actor and critic networks. Each episode terminates either when the number of steps reaches the time limit T, or when the system state has reached an inadmissible and non-reversible value.

Algorithm 1 DDPG algorithm

Randomly initialize critic network $\mathcal{C}(s, a|\theta^{\mathcal{C}})$ and actor $\mathcal{A}(s|\theta^{\mathcal{A}})$ with weights $\theta^{\mathcal{C}}$ and $\theta^{\mathcal{A}}$
Initialize target network $\mathcal{C}'$ and $\mathcal{A}'$ with $\theta^{\mathcal{C}'} \leftarrow \theta^{\mathcal{C}}$ and $\theta^{\mathcal{A}'} \leftarrow \theta^{\mathcal{A}}$
Initialize eplay buffer R
for $e = 1, 2, \ldots, E$ **do**
    Define the random process noise $\mathcal{N}_e$ for action exploration
    Receive initial observation state s_1
    for $t = 1, 2, \ldots, T$ **do**
        Select action $a_t = \mathcal{A}(s_t|\theta^{\mathcal{A}}) + \mathcal{N}_e$
        Execute $a_t = \mathcal{A}$ and observe r_t and s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in R
        Sample N-transitions random minibatch from R
        Set $y_i = r_i + \gamma \mathcal{C}'(s_{i+1}, \mathcal{A}'(s_{i+1}|\theta^{\mathcal{A}'})|\theta^{\mathcal{C}'})$
        Update critic minimizing the loss:
        $L = \frac{1}{N}\sum_i (y_i - \mathcal{C}(s_i, a_i|\theta^{\mathcal{C}}))^2$
        Update actor policy using the policy gradient:
        $\nabla_{\theta^{\mathcal{A}}} J \approx \frac{1}{N}\sum_i \nabla_a \mathcal{C}(s, a|\theta^{\mathcal{C}})|_{s_i, \mathcal{A}(s_i)} \nabla_{\theta^{\mathcal{A}}} \mathcal{A}(s, |\theta^{\mathcal{A}})|_{s_i}$
        Update target networks, with $\tau \ll 1$:
        $\theta^{\mathcal{C}'} \leftarrow \tau\theta^{\mathcal{C}} + (1-\tau)\theta^{\mathcal{C}'}$
        $\theta^{\mathcal{A}'} \leftarrow \tau\theta^{\mathcal{A}} + (1-\tau)\theta^{\mathcal{A}'}$
    end for
end for

At the end of the training procedure, the trained DDPG agent can be deployed in the so-called exploitation phase: the agent is tested over the environment, without inserting any process noise to the actor network output layer.

III. System Mathematical Modeling

The Bergman Model is the simplest model that effectively describes the blood glucose-insulin regulatory system [21]. It is characterized by the following equations:

$$\begin{cases} \dot{G} &= -p_1[G - G_b] - GX + d \\ \dot{X} &= -p_2 X + p_3[I - I_b] \\ \dot{I} &= -n[I - I_b] + \delta[G - h]^+ t + u \end{cases}, \tag{2}$$

where $[G-h]^+ := \max(G-h, 0)$.

The variables and parameters of the model are illustrated in Table I and Table II, respectively.

Note that both the control input signal $u(t)$ and the measurable disturbance signal $d(t)$ are non-negative.
The equilibrium state of the dynamic system is given by:

$$x_e = \begin{bmatrix} G_b \\ 0 \\ I_b \end{bmatrix}. \tag{3}$$

A change of coordinates is performed in order to center such equilibrium state in $\mathbf{0}$:

$$z = x - x_e = \begin{bmatrix} \tilde{G} \\ \tilde{X} \\ \tilde{I} \end{bmatrix} = \begin{bmatrix} G - G_b \\ X - 0 \\ I - I_b \end{bmatrix}, \tag{4}$$

so that:

$$z_e = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \dot{z} = \begin{bmatrix} -p_1\tilde{G} - (\tilde{G} + G_b)\tilde{X} + d \\ -p_2\tilde{X} + p_3\tilde{I} \\ -n\tilde{I} + \delta[\tilde{G} + G_b - h]^+ t + u \end{bmatrix}. \tag{5}$$

Note that $\delta = 0$ for type 1 diabetic patients.

A physiological constraint on the amount of injected insulin is imposed in order to limitate the risk of causing hypoglycemic episodes [22]:

$$0 \leq u(t) \leq 4.27 \ \mu\text{U/min}. \tag{6}$$

This system translates to the MDP framework in the following way: the state space is $\mathcal{S} = \{(\tilde{G}, d)\}$, the action space is $\mathcal{A} = \{u | u \in [0, 4.27] \ \mu\text{U/min}\}$, and the reward function is:

$$r(G) = \begin{cases} 1, & \text{if } 70 \leq G \leq 120, \\ 0.3, & \text{if } 120 < G \leq 160, \\ 0.1, & \text{if } 50 \leq G < 70 \\ -0.4 - \frac{(G-160)}{200}, & \text{if } 160 < G \leq 280, \\ -0.6 - \frac{(G-50)}{100}, & \text{if } 10 \leq G < 50, \\ -1, & \text{otherwise.} \end{cases} \tag{7}$$

Note that low positive rewards are given when blood glucose concentration approaches the limits of normoglycemia. Moreover, hypoglycemia is penalized slightly more than hyperglycemia, since it yields worse clinical consequences.

IV. Simulations and Results

In order to evaluate the performances of the proposed method, simulations of its effectiveness were performed on the Bergman glucoregulatory system presented in Section III in a Python 3.10 environment using Tensorflow and Keras, and a NVIDIA Tesla T4 GPU.

Disturbance is simulated as in [23]: it is assumed that throughout a day three meals are consumed, cointaining 60, 110 and 90 grams of carbohydrates respectively. On such nominal values a gaussian noise with mean $\mu = 0$ and standard deviation $\sigma = 0.2$ is added during training of the DDPG model in order to simulate the variability that is proper of realistic scenarios. The numerical values of the model parameters in Table II are given as follows: $G_b = 75$, $I_b = 15$, $p_1 = 2.3 \times 10^{-6}$, $p_2 = 0.088$, $p_3 = 0.63 \times 10^{-3}$, $\delta = 0$, $n = 0.09$.

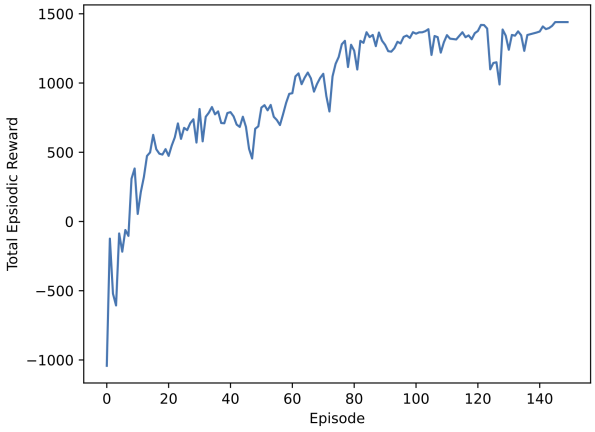

Fig. 4. Evolution of the cumulative reward per episode over the training episodes.

Since RL algorithms only deal with discrete-time dynamics, the equations in (5) are discretized via the fourth-order Runge-Kutta method, with time step $dt = 1$ [min].

The hyperparameters of the DDPG algorithm are carefully selected as follows: a discount factor of value $\gamma = 0.9$, an actor learning rate of value $\alpha = 0.0001$, and a critic learning rate of value $\beta = 0.0005$. Both the actor and critic share the same neural architecture composed of three layers, the first two of 512 and 128 neurons with ReLU [24] activation function, and the last one of one neuron only, with hyperbolic tangent activation function for the actor.

The DDPG agent is trained for $E = 150$ episodes, each one lasting $T = 1440$ [min], thus for a total of 1440 steps (T/dt).

Figure 4 shows the evolution of the cumulative reward per episode over the training episodes, converging to the maximum feasible value given the reward design illustrated in (7).

Figures 5 and 6 illustrate respectively the daily profile

TABLE I
Variables of the Bergman Model

Variable	Description	Unit
G	Plasma glucose concentration	mg/dl
X	Insulin's effect on net glucose disappearance	min^{-1}
I	Insulin concentration in plasma	$\mu\text{U/ml}$
u	Insulin infusion rate	$\frac{\mu\text{U}}{\text{ml min}}$
d	Measurable meal disturbance	$\frac{\text{mg}}{\text{dl min}}$

TABLE II
Parameters of the Bergman Model

Variable	Description	Unit
G_b	Basal glucose plasma concentration	mg/dl
I_b	Basal insulin plasma concentration	$\mu\text{U/ml}$
p_1	Indep. glucose disappearance rate	min^{-1}
p_2	Spontaneous glucose uptake ability rate	min^{-1}
p_3	Insulin-dep. glucose uptake ability rate	$\frac{\text{ml}}{\mu\text{U min}^2}$
δ	Rate of pancreatic β-cells insulin release	$\frac{\mu\text{U dl}}{\text{ml mg min}^2}$
n	Insulin infusion rate	min^{-1}

of the blood glucose concentration and of the control effort enforced by the corresponding trained model, as well as by the other control logics illustrated in the following. Note that the trained DDPG controller successfully maintains the blood glucose concentration in the normoglycemic range (green area), avoiding both hyperglycemic episodes (yellow area) and hypoglycemic episodes (red area).

In order to assess the performances of the proposed method, benchmark comparisons with other control logics were conducted.

A first comparison is performed with respect to standard clinical practice (CP) as in [25], taking the following expression:

$$u_t = \frac{\text{CHO}_t}{\text{ICRR}} + \frac{G_t - G_{ref}}{\text{ISF}} - \text{IOB}_{t-1}, \tag{8}$$

where u_t is the insulin injected at time t, CHO_t is the carbohydrates intake at time t, ICR is the insulin-to-carbohydrate ratio, ICR is the insulin sensitivity factor, G_t is the blood glucose concentration at time t, G_{ref} is its reference value, and IOB_{t-1} is the insulin on board at previous time $t-1$.

Figures 5 and 6 again depict respectively the daily profile of the blood glucose concentration and of the control input enforced by standard clinical practice.

A second comparison is performed using Proportional Integral Derivative (PID) control [26], where the input signal is given by:

$$u_t = K_P e_t + K_I \int_0^t e_\tau \, d\tau \; + K_D \frac{de_t}{dt}, \tag{9}$$

where $e_t = G - G_{ref}$. The control gains are tuned by means of empirical evidence in order to achieve a satisfactory trade-off between robustness and transient speed, resulting in the following values: $K_P = -0.001$, $K_I = -7 \times 10^{-6}$, $K_D = 0.015$.

The resulting profiles of the blood glucose concentration and of the control effort enforced by the PID controller are shown again in Figures 5 and 6 respectively.

Also in this case an hyperglycemic episode occurs during the day. Moreover note that, since PID tracks a reference by minimizing error without including bounds to avoid hyperglycemia and hypoglycemia, higher peaks of the blood glucose concentration are present.

Lastly, a final comparison is performed with respect to Nonlinear Model Predictive Control (NMPC), an optimization-based control strategy that relies on solving a finite-horizon optimization problem at each time step to determine the control input by minimizing an objective function while satisfying system dynamics and state and input constraints.

Figures 5 and 6 again depict respectively the daily profile of the blood glucose concentration and of the control input enforced by NMPC.

Although the NMPC controller, analogously to the DDPG controller, effectively prevents both hyperglycemic and hypoglycemic episodes, the latter presents some advantages with respect to the former. First, performing inference with the DDPG controller is much faster than solving the NMPC optimization problem; although with a sampling time of 1 min such difference results negligible, it is expected that in future works, using more complex glucoregulatory models, the superior speed of the DDPG contorller makes it more suitable for real-time applications with respect to NMPC. Moreover, the peaks of the NMPC control effort reach much higher values with respect to the DDPG controller, reaching saturation at the maximum limit imposed by the physiological constraint on the amount of administrable insulin given by (6). The clinical relevance of lowering the maximum control effort peak consists of minimizing the discomfort of the patient, reducing the risk of hypoglycemia, and improving the overall glycemic control since rapid drops and increases in blood glucose concentration levels can be destabilizing.

Table III synthesizes the Key Performance Indicators (KPIs) that are useful to compare the outcomes of the four presented control logics.

V. Conclusions

This work produced a novel control logic for autonomous glycemic control of type 1 diabetes based on Deep Deterministic Policy Gradient. The proposed method has been vali-

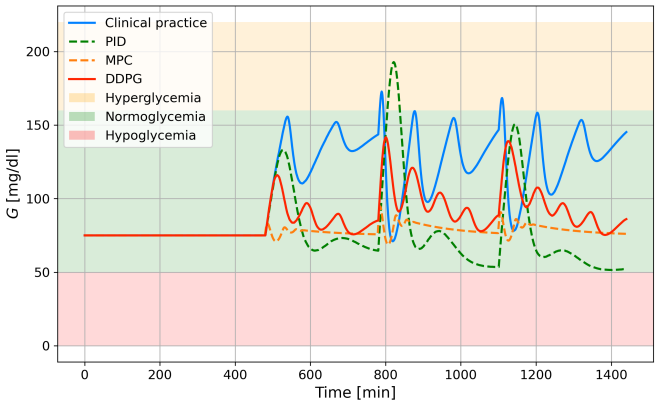

Fig. 5. Profile of the daily blood glucose concentration obtained with the analyzed control logics: clinical practice (blue), PID (green), MPC (yellow), and DDPG (red).

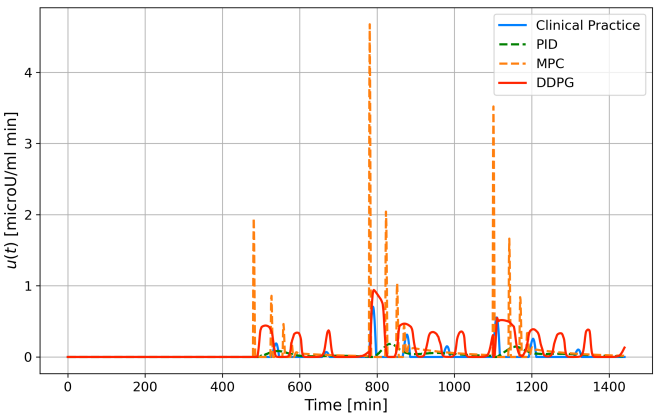

Fig. 6. Daily evolution of the control effort enforced by the analyzed control logics: clinical pratice (blue), PID (green), MPC (yellow), and DDPG (red).

TABLE III

KPIs COMPARISON OF THE USED CONTROL LOGICS

KPI	CP	PID	NMPC	DDPG
Minimum glycemic value [mg/dl]	71	51	75	75
Maximum glycemic value [mg/dl]	172	192	116	142
Time in hypoglycemia [min]	0	0	0	0
Time in hyperglycemia [min]	20	33	0	0
Total injected insulin [μU]	27.6	41.5	89.4	93.6
Computing time [sec]	1.7×10^{-6}	2.9×10^{-6}	2.4×10^{-2}	2.1×10^{-3}
Maximum control effort peak [μU]	0.69	0.19	4.68	0.94

dated through in-vitro simulations on the Bergman glucoregulatory model, and compared with standard clinical practice, Proportional Integral Derivative control, and Nonlinear Model Predictive Control. The key research objective of this work has been accomplished: the proposed control logic shows superior performances with respect to the others in terms of insulin regulation, successfully preventing episodes of both hyperglycemia and hypoglycemia, while reducing control effort peaks and computation time.

Future works shall tackle the limitations of this study: findings are related to the Bergman minimal model, which is among the simplest ones for describing Type 1 diabetes, thus more complex glucoregulatory models will be used in order to describe more realistic scenarios in daily life of diabetic patients, with the introduction of additional disturbances such as stress and physical activity; moreover, the KPIs that have been analyzed in Table III will be used as additional reward function terms, in order to further improve the performances of the trained model.

REFERENCES

[1] World Health Organization, "Diabetes." https://www.who.int/en/news-room/fact-sheets/detail/diabetes, 2023. Accessed: May 18th, 2023.

[2] M. A. Atkinson, G. S. Eisenbarth, and A. W. Michels, "Type 1 diabetes," *The Lancet*, vol. 383, no. 9911, pp. 69–82, 2014.

[3] E. Bekiari, K. Kitsios, H. Thabit, M. Tauschmann, E. Athanasiadou, T. Karagiannis, A.-B. Haidich, R. Hovorka, and A. Tsapas, "Artificial pancreas treatment for outpatients with type 1 diabetes: systematic review and meta-analysis," *bmj*, vol. 361, 2018.

[4] F. Chee, T. L. Fernando, A. V. Savkin, and V. Van Heeden, "Expert pid control system for blood glucose control in critically ill patients," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 4, pp. 419–425, 2003.

[5] C. Li and R. Hu, "Pid control based on bp neural network for the regulation of blood glucose level in diabetes," in *2007 IEEE 7Th international symposium on bioinformatics and bioengineering*, pp. 1168–1172, IEEE, 2007.

[6] D. U. Campos-Delgado, M. Hernández-Ordoñez, R. Femat, and A. Gordillo-Moscoso, "Fuzzy-based controller for glucose regulation in type-1 diabetic patients by subcutaneous route," *IEEE Transactions on biomedical engineering*, vol. 53, no. 11, pp. 2201–2210, 2006.

[7] P. H. Colmegna, R. S. Sanchez-Pena, R. Gondhalekar, E. Dassau, and F. J. Doyle, "Switched lpv glucose control in type 1 diabetes," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1192–1200, 2015.

[8] Q. Wang, S. Harsh, P. Molenaar, and K. Freeman, "Developing personalized empirical models for type-i diabetes: An extended kalman filter approach," in *2013 American Control Conference*, pp. 2923–2928, IEEE, 2013.

[9] H. Lee and B. W. Bequette, "A closed-loop artificial pancreas based on model predictive control: Human-friendly identification and automatic meal disturbance rejection," *Biomedical Signal Processing and Control*, vol. 4, no. 4, pp. 347–354, 2009.

[10] L. Magni, D. M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli, "Model predictive control of type 1 diabetes: an in silico trial," 2007.

[11] S. D. Patek, B. W. Bequette, M. Breton, B. A. Buckingham, E. Dassau, F. J. Doyle III, J. Lum, L. Magni, and H. Zisser, "In silico preclinical trials: methodology and engineering guide to closed-loop control in type 1 diabetes mellitus," *Journal of diabetes science and technology*, vol. 3, no. 2, pp. 269–282, 2009.

[12] D. Sui, L. Feng, and M. Hovd, "Algorithms for online implementations of explicit mpc solutions," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3619–3622, 2008.

[13] S. Bahremand, H. S. Ko, R. Balouchzadeh, H. Felix Lee, S. Park, and G. Kwon, "Neural network-based model predictive control for type 1 diabetic rats on artificial pancreas system," *Medical & biological engineering & computing*, vol. 57, pp. 177–191, 2019.

[14] M. Tejedor, A. Z. Woldaregay, and F. Godtliebsen, "Reinforcement learning application in diabetes blood glucose control: A systematic review," *Artificial intelligence in medicine*, vol. 104, p. 101836, 2020.

[15] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1223–1232, 2020.

[16] T. Zhu, K. Li, L. Kuang, P. Herrero, and P. Georgiou, "An insulin bolus advisor for type 1 diabetes using deep reinforcement learning," *Sensors*, vol. 20, no. 18, p. 5058, 2020.

[17] K.-S. Hwang, Y.-J. Chen, T.-F. Lin, and W.-C. Jiang, "Continuous action for multi-agent q-learning," in *2011 8th Asian Control Conference (ASCC)*, pp. 418–423, IEEE, 2011.

[18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[19] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[20] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, Pmlr, 2014.

[21] A. G. Gallardo-Hernández, M. A. González-Olvera, M. Castellanos-Fuentes, J. Escobar, C. Revilla-Monsalve, A. L. Hernandez-Perez, and R. Leder, "Minimally-invasive and efficient method to accurately fit the bergman minimal model to diabetes type 2," *Cellular and Molecular Bioengineering*, vol. 15, no. 3, pp. 267–279, 2022.

[22] T. Heise, E. Zijlstra, L. Nosek, T. Rikte, and H. Haahr, "Pharmacological properties of faster-acting insulin aspart vs insulin aspart in patients with type 1 diabetes receiving continuous subcutaneous insulin infusion: a randomized, double-blind, crossover trial," *Diabetes, Obesity and Metabolism*, vol. 19, no. 2, pp. 208–215, 2017.

[23] C. Dalla Man, R. A. Rizza, and C. Cobelli, "Meal simulation model of the glucose-insulin system," *IEEE Transactions on biomedical engineering*, vol. 54, no. 10, pp. 1740–1749, 2007.

[24] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[25] H. Zisser, L. Robinson, W. Bevier, E. Dassau, C. Ellingsen, F. J. Doyle III, and L. Jovanovic, "Bolus calculator: a review of four "smart" insulin pumps," *Diabetes technology & therapeutics*, vol. 10, no. 6, pp. 441–444, 2008.

[26] M. Goharimanesh, A. Lashkaripour, and A. Abouei Mehrizi, "Fractional order pid controller for diabetes patients," *Journal of Computational Applied Mechanics*, vol. 46, no. 1, pp. 69–76, 2015.