Long-Term Adaptation of Closed-Loop Glucose Regulation Via Reinforcement Learning Tools

María Cecilia Serafini [*,**] **Nicolás Rosales** [*] **Fabricio Garelli** [*]

[*] *Grupo de Control Aplicado (GCA). LEICI institute, Facultad de Ingenieria (UNLP-CONICET) (email:cecilia.serafini@ing.unlp.edu.ar)*
[**] *Comisión de Investigaciones Científicas (CICpBA)*

Abstract: In recent years, closed-loop controllers for glucose regulation, also called Artificial Pancreas (AP) systems, have become an emblematic problem in the field of automatic control. Several closed-loop systems are in development around the world, such is the case of the Automatic Regulation of Glucose (ARG) algorithm developed in Argentina that has already been tested in the first AP clinical trials for all Latin America. Due to the complexity of the problem at hand, the design and evaluation of controllers for glucose regulation is mostly centered around short-term performance, especially focusing on postprandial periods. However, as everybody, people with diabetes undergo changes in their routines or physiology that can result in an inadequate performance if the controller is not adapted correctly.
In this work, the potential of Reinforcement Learning (RL) tools for long-term adaptation of the ARG controller is evaluated through a discrete Q-learning agent. The proposed strategy is evaluated *in-silico* using the UVA simulator, modifying only one parameter of the controller: its Insulin-On-Board maximum limit. Results show that RL successfully adapts the controller avoiding hypoglycemia when the subjects' physiology changes through time, and that the trained agent outperforms a rule-based decision making scheme for the majority of the adult population.

Keywords: Diabetes Mellitus, Closed-Loop, Reinforcement Learning

1. INTRODUCTION

Type 1 Diabetes Mellitus (T1DM) is one of the most burdening health conditions worldwide (Vos and Lim [2020]). It is an autoimmune metabolic disorder which alters blood glucose regulation when the β cells in the pancreas are destroyed and this organ reduces or completely looses the ability to produce insulin. Without insulin, elevated levels of glucose remain in the bloodstream, generating hyperglycemia which in the long run is associated with severe micro and macro-vascular damage.

People who live with T1DM usually depend on exogenous insulin analogues injection. This synthetic hormone can be administered manually (MDI- Multiple Daily Injections) or through Continuous Subcutaneous Insulin Infusion (CSII) systems, also known as insulin pumps. The development of the latter and Continuous Glucose Monitoring (CGM) devices has lead the incorporation of control algorithms to automatize insulin infusion. Both hybrid and fully closed-loop systems are commonly known as Artificial Pancreas (AP) systems and different strategies are in development around the globe (Lanzola et al. [2015], Haidar [2016], Fushimi et al. [2020], Rosales et al. [2022]).

Each diabetic person has their own particular metabolic process and might respond differently to the same treatment (inter-patient variability) and can also present variations within themselves (intra-patient variability) and need personalized treatments that vary according to their bodies and lifestyles, not only considering intra-day variations but also inter-day and long-term ones (Ruan et al. [2017]). This makes controller adaptation a necessary subject to adjust the treatment according to the changes in every subject. Some work has been done, especially for a day-to-day adaptation (Messori et al. [2017]), but the problem of systematic adaptation while maintaining AP performance is far from resolved (Toffanin et al. [2018]).

Reinforcement Learning (RL) is a particular branch of Machine Learning (ML) that is considered to be in-between supervised and unsupervised learning (Sutton and Barto [1998]), in which an agent learns by interacting iteratively with a system or environment. Given a certain state of the system and an action chosen by the agent, the environment evolves to a different state and returns a reward that indicates whether that action was good (or bad) in the context of the specific scenario. RL has an extensive theoretical background, and has been studied since the 1980s but has only recently begun to be tested in practical applications thanks to the advances of technology regarding computing power (Nian et al. [2020]).

As a consequence of the "trial-and-error" nature of RL, many applications require the initial training of the agents to be conducted over complex simulated models so as to ensure safety of people and equipment. In the medical field, RL has mostly been used for prognosis, classification and diagnosis, making use of the big amounts of data generated by health systems, but only a few clinical trials use this tool (Oroojeni Mohammad Javad et al. [2019]). Such is the case that only in 2019, a commentary on guides for working with RL in medicine was published by Nature (Gottesman et al. [2019]).

In the AP area, specifically, there is some work using RL agents as controllers for glycemic control, the most used configuration is that of the Actor-Critic (Daskalaki et al. [2016], Sun et al. [2019]) and Gaussian Processes (De Paula et al. [2015]). Agents derived from Q-Learning algorithm, a discrete-space RL tool characterized by its simplicity and relative ease of implementation, have also been used. This type of agent has been mostly used as a direct replacement of the controller (Tejedor et al. [2020], Ngo et al. [2018], Fox and Wiens [2019]). With this configuration, the control algorithm becomes a black box, and thus the rigorousness and advantages of automatic control theory to robustly deal with nonlinear and uncertain systems can not be fully exploited.

In this work, a Q-learning based adaptation technique for the Automatic Regulation of Glucose (ARG) algorithm (Sánchez-Peña et al. [2018]) is developed, considering long-term ongoing non-predictable Insulin Sensitivity (IS) variation. The developed strategy modifies only one parameter in the chosen AP system (the Insulin on Board (IOB) limit) instead of replacing the controller entirely. The proposal aims at adapting glucose control systems in the long run, considering extended periods of time and not only immediate changes but also possible future scenarios which is where RL has its strength, since it has the possibility of taking probable future states into account. To this end, the discount factor of the Q-learning algorithm is designed to consider long-term rewards instead of focusing primarily on short-term results, simulations steps are proposed as week intervals and training episodes consist of approximately 4 months. The proposed adaptation strategy is tested *in-silico* on the adult cohort of the UVA/Padova simulator Dalla Man et al. [2014].

The structure of this paper is as follows: Section 2 presents a summary of RL and its uses in AP systems, section 3 describes the ARG algorithm, the experiment design and the proposed strategy, section 4 shows the results for the proposed strategy in the adult population of the UVA/Padova. The *in-silico* results are compared to a manual adaptation scheme. Lastly, in sections 5 the results are discussed and future lines of work are presented.

2. REINFORCEMENT LEARNING

Presented in Sutton and Barto [1998] and extended by the same and other authors since then (Szepesvári [2010], Sutton and Barto [2018]), the basic structure of an RL system is shown in Fig. 1. It consists of an agent that interacts iteratively with a given environment and, through trial-and-error, "learns" the best action for each state the system is in. Every time the agent takes an action a, this action modifies the environment, and it transitions form state s to state s' and gets a scalar reward r.

The goal of the agent is to maximize the reward received in any state and obtain an optimal state-action map called a **policy** $\pi(a,s)$ that maximizes the sum of all rewards, called a value function $V_\pi(s)$, associated with a given policy π:

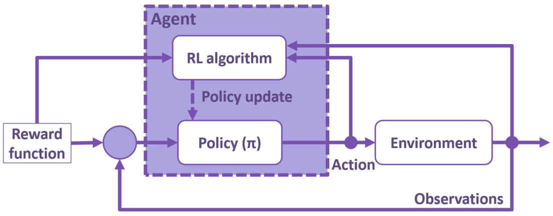

Figure 1. Basic structure of a Reinforcement Learning System.

$$V_\pi(s) = E\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3+\ldots}\right\} \quad (1)$$

where $\gamma \in [0,1)$ is called a **discount factor** and it determines the weight of future rewards. If $\gamma = 0$ then all possible future reward will be ignored and if $\gamma \approx 1$ then the agent will give future rewards the same importance as short term ones.

Optimal control theory postulates, through Bellman's optimality Equation, that there is a necessary condition for the policy to be optimal, this is defined as following:

$$V_*(s) = \max_a \sum p(s', r|s,a)[r + \gamma V_*(s')] \quad (2)$$

where $p(s', r|s,a)$ is the transition probability from state s to state s' when taking action a. In most practical cases, Bellman's optimality equation cannot be solved directly, but Sutton and Barto consider that RL algorithms are approximate solutions to it (Sutton and Barto [2018]).

In RL's basic structure, the controller is completely replaced by the policy learnt by the agent. This means that the controller becomes a black-box, which can be problematic. This issue can be bypassed considering the controller as part of the environment and letting the agent interact with it generating a policy that, instead of returning a control action directly, acts modifying one or more parameters of the controller itself. Such a structure can be seen in Fig. 2. This approach has been used mostly in the field of robotics, to adapt PID controllers by modifying their gains (Carlucho et al. [2019]). In the AP area in particular, RL has mostly been used following the structure shown in Fig. 1 and specially focused on short term and postprandial control (Tejedor et al. [2020]).

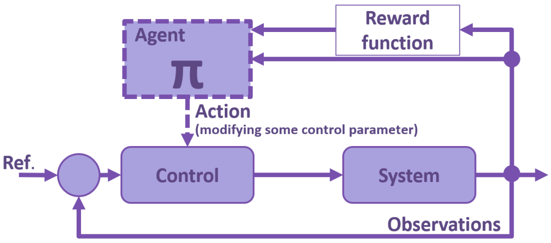

Figure 2. Structure of a Reinforcement Learning System for controller adaptation.

Taking all of the information summarized in this section into consideration, an approach adapting previously designed controllers instead of replacing them is adopted here to further exploit control theory developments and, also, to allow building on top of any previous work.

3. Q-LEARNING FOR LONG-TERM AP ADAPTATION

There are different approaches for RL problems depending on the characteristics of the system and computing resources. One of them is Q-Learning, which uses a parameterized version of the value function, called the Q-value function $Q(s, a)$, that is maximized by π.

3.1 General Q-Learning concepts

Q-learning is a RL algorithm initially developed by Watkins [1989] and it is considered one of the major breakthroughs in Temporal-Difference Learning. Its basic structure is shown in Alg. 1.

In this work, a self developed Q-Learning code, oriented to long-term adaptation and based on closed-loop strategies for glycemic control is implemented considering a tabular policy and discrete action and state spaces. Developing the Q-learning algorithm from scratch yields the advantage of being able to tweak every parameter to adjust it for the particular problem at hand. This is significant given the fact that this approach does not replace the controller but only modifies some parameters, and thus the use of prefabricated toolboxes does not directly apply.

For the following work, take into account that a "state" in RL is not necessarily a system state variable as understood in control theory but a collection of observations.

Algorithm 1: Basic Q-Learning algorithm.

```
Algorithm parameters: Learning rate $\alpha \in (0, 1]$, $\varepsilon > 0$, $\gamma \in [0, 1)$
Initialize $Q(s, a)$ for all $s \in S$, $a \in A$. $Q(terminal, :) = 0$
Repeat for each episode
    Initialize $s \in S$
    Repeat for each step
        Choose $a \in A$ for the current state $s$ through $Q(s, a)$
            following $\varepsilon$-greedy policy (see Alg. 2)
        Take action $a$, observe reward $r$ and next state $s'$
        Get maximizing action from $s'$ : $\max_{a \in A} Q(s', A)$
        Update $Q$: $Q(s, a) \leftarrow Q(s, a)$
            $+\alpha[r + \gamma \cdot \max_{a \in A} Q(s', A) - Q(s, a)]$
        $s \leftarrow s'$
    until $s$ is terminal (end of episode)
until last episode
```

For the problem at hand, every step was considered to be the result after one week of simulation. The episodes consisted of 16 steps and the states were of the form $s = (s_1, s_2)$, where s_1, s_2 represent the discretized percentages of time in hypoglycemia (Glucose value < 70 mg/dl) and hyperglycemia (Glucose value > 180 mg/dl), respectively, calculated after a full step, looking at the resulting glucose vector.

As was shown in the previous section and in Alg. 1, one of the main parameters of Q-Learning is γ, the discount factor that defines whether the system has a "short-term mind" or a long-term one. In this particular case, it was necessary to take future possible states into account, as much as immediate results. Considering this, the γ parameter was chosen to be near 1.

Another important factor to consider in RL is the balance between exploration (testing all the possible actions and their outcomes in every state) and exploitation (maximizing the reward). To reduce action exploration gradually, an epsilon-greedy policy with decaying epsilon was implemented (see Alg. 2). To allow for the action exploration to be reduced over time, ε was defined as:

$$\varepsilon = N_0/(N_0 + N(s)) \tag{3}$$

where $N(s)$ represents the number of visits to a given state s, and N_0 is a fixed value so that the value of ϵ is state-dependent and inversely proportional to the number of visits to a given state.

On the other hand, to avoid that the most frequent actions change Q values disproportionately and to improve convergence efficiency, a state/action dependent and decaying learning rate was implemented using the rule

$$\alpha = 1/N(s, a) \tag{4}$$

where $N(s, a)$ represents the number of times that a given action a has been taken from state s.

Algorithm 2: *ε-greedy* strategy with decaying ε.

```
Define $N_0$ and initialize $N(s)$
For each visited state $s$
    $N(s) \leftarrow N(s) + 1$
    $\varepsilon = N_0/(N_0 + N(s))$
    if $rand < \varepsilon$ then
        take uniformly random action (probability $\varepsilon$)
    else
        take action $a$ that maximizes $Q$ from $s$:
          $= \max_{a \in A} Q(s', A)$
    end
end
```

A summary of all parameters and configurations for the present work can be found in subsection 3.3

3.2 ARG algorithm: Automatic Regulation of Glucose

The ARG algorithm was developed and clinically tested in Argentina. It's block diagram can be seen in Fig. 3.

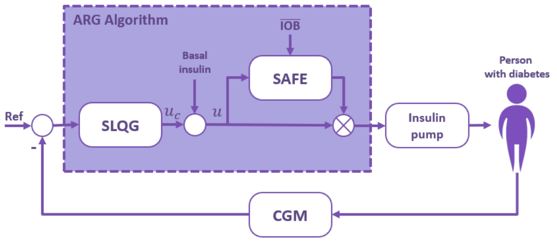

Figure 3. Block diagram for the ARG algorithm.

The algorithm is non-hybrid, since no premeal boluses are infused, and it is based on a main Switched Linear Quadratic Gaussian (SLQG) controller that has the patient's open-loop basal insulin added. The SLQG switches between an aggressive mode, replacing the open-loop bolus to compensate for meals, and a conservative mode to keep the patient in normoglycemia (glucose value $\in [70 - 180]$ mg/dl) during fasting periods.

The dose of insulin indicated by the SLQG is multiplied by the output signal of the Safety Auxiliary Feedback Element (SAFE) block, whose function is to modulate the insulin infusion by imposing a restriction on the IOB estimation to avoid possible hypoglycemia. The IOB limit

($\overline{\text{IOB}}$) is considered in this case as a piecewise constant function, defined according to a meal classification. This limit is the parameter that was chosen to be modified with the actions taken by the RL agent. For a more detailed description of the algorithm, see Colmegna et al. [2018].

3.3 Experiment design

In order to test the controller's ability to adapt and learn from long-term variations, the experiment was designed considering variations in IS, as this parameter strongly affects how every person responds to treatment.

This was implemented considering no variation for the first 6 simulation steps and random proportional changes on the remaning 10 steps (with possible values $\{+10\%; -10\%; 0\}$), applied to the parameters V_{mx} and K_{P3} of the UVA models. These parameters represent the peripheral tissue and hepatic insulin sensitivity, respectively (see Dalla Man et al. [2014] for the complete set of model equations).

The simulation scenario for the training phase was set as following: 10 adults with IS variation as defined previously and 10 hours of simulation, with one meal after one hour, which allows the analysis of the whole postprandial stage. Since the agent only receives data of percentage of time in hyper and hypoglycemia from the past step, this interval of simulation could represent any given amount of time after which one would like to take an action. In this case it was considered to represent the average response to one week of treatment. Then, each episode consistent of 16 steps can be considered as approximately 4 months. This interpretation allows reducing the computation time for the training phase, and also considers the fact that it is unlikely for a patient to change their insulin calculations on a daily bases but more likely to modify them weekly.

The limit $\overline{\text{IOB}}$ was chosen to be modified via RL algorithm, using a proportional gain K to adapt it for each individual separately, considering $K = 1$ as the standard ARG controller. Then,

$$\overline{\text{IOB}}_{adapt} = K * \overline{\text{IOB}} \tag{5}$$

where $\overline{\text{IOB}}_{adapt}$ is the limit obtained after applying the optimal action according to the policy and $\overline{\text{IOB}}$ is the original limit of the ARG algorithm.

The summary of RL training parameters is as follows:

- Episode: 16 steps, representing approximately 4 months of treatment.
- Simulation step: 1 full simulation of 10 hours, including one meal and its full postprandial period, representing the average glucose response after one week of treatment. IS modified applying random variation to V_{mx} and K_{p3} in every simulation step.
- State space: $s \in S$, where every s is of the form $s = (s_1, s_2)$, and s_1, s_2, discretized percentages of time in hypoglycemia and hyperglycemia, respectively.
- Action space: $A = [0\%, \pm1\%, \pm2\%, \pm3\%, \pm4\% \pm 5\% \pm 10\%]$ applied directly over K, the proportional gain that modifies the $\overline{\text{IOB}}$.
- Reward function:

$$r = \begin{cases} -2 & \text{if } s_1 > hypo_{max} \\ -1 & \text{if } s_1 \leq hypo_{max} \\ & \text{and } s_2 > hyper_{max} \\ +10 & \text{otherwise} \end{cases} \tag{6}$$

  where $hypo_{max}$ and $hyper_{max}$ are the the maximum accepted values for s_1 and s_2, respectively.
- $\gamma = 0.9$, ε and α as defined in Eqs. (3) and (4)
- Discretization schemes
  · s_1 (% of time in hypoglycemia) was discretized as: [0 | 1 | 2.5 | 5 | 10 | 12 | 15 | 20 | 25]
  and $hypo_{max}$ was chosen to be $2,5\%$
  · s_2 (% of time in hyperglycemia) was discretized as: [0 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50]
  and $hyper_{max}$ was chosen to be 20%

State discretization was carried out manually into the intervals shown above and limits were chosen taking the last consensus for clinical results into account (Battelino et al. [2019]) and defining even more restricting conditions than the required metrics. The reward function was defined using a similar criterion as the one followed by most benchmark RL training problems, considering for this particular case first hypoglycemia and secondly hyperglycemia.

The training process was carried out through 70000 simulations, 7000 for each adult patient, with a common policy for all of them.

3.4 Testing

After training, the agent was tested with a modified and more challenging scenario, with IS varying randomly ±20% or 0%, throughout all 16 steps. This simulates an unstable patient, such as someone with labile diabetes, as reported in previous clinical trials carried out by the present work group. An example of such variation can be seen in Fig. 4.

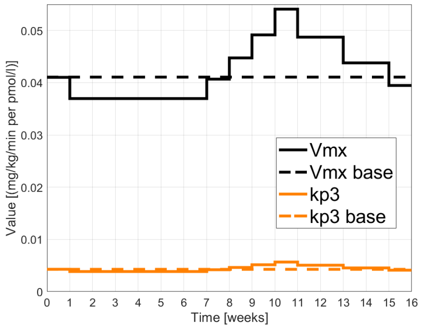

Figure 4. Variations in V_{mx} and K_{p3} (solid lines) from their nominal - base - values are (dashed line).

The results of adaptation via the RL policy application were compared to a manual adaptation scheme with the same level of complexity as the reward function, using a fixed action of ±5% designed as shown in Eq. (7).

$$K = \begin{cases} K * (1 - 5/100) & \text{if } s_1 > hypo_{max} \\ K * (1 + 5/100) & \text{if } s_1 \leq hypo_{max} \\ & \text{and } s_2 > hyper_{max} \\ K & \text{otherwise} \end{cases} \tag{7}$$

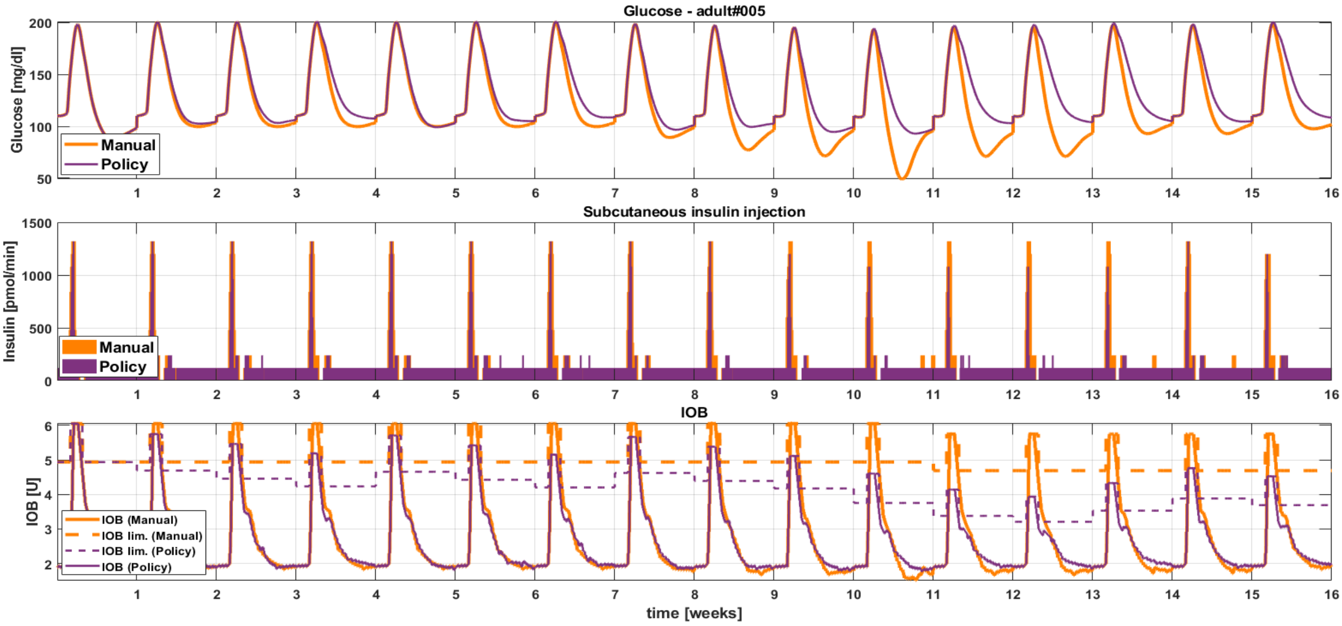

Figure 5. Glucose excursion (top), insulin infusion (mid) and IOB (bottom) evolution over time for Adult#05 using the ARG algorithm with RL policy application (purple thinner) and with manual scheme (orange thicker). At bottom: IOB (solid line) and $\overline{\text{IOB}}$ (dashed line).

Table 1. Comparison between strategies for adaptation of the IOB limit. Percentage of time in hyper- and hypoglycemia considering the full 16 weeks, and maximum % reached.

	Manual strategy with action ±5%						Policy application					
Subject	% Time	% Time	% T. hypo	% Time	% Time	% T. hyper	% Time	% Time	% T. hypo	% Time	% Time	% T. hyper
	↓ 54 mg/dL	↓ 70 mg/dL	max value	↑180 mg/dL	↑250 mg/dL	max value	↓ 54 mg/dL	↓ 70 mg/dL	max value	↑180 mg/dL	↑250 mg/dL	max value
adult#01	0,00	0,68	7,65	9,74	0,00	11,48	0,00	0,00	0,00	13,29	0,00	17,97
adult#02	0,00	0,39	6,16	4,01	0,00	6,99	0,00	0,00	0,00	13,54	0,00	33,78
adult#03	0,00	1,36	21,80	4,58	0,00	8,49	0,00	0,00	0,00	18,89	0,00	77,20
adult#04	0,00	1,46	16,47	4,67	0,00	7,32	0,00	0,00	0,00	12,86	0,00	25,79
adult#05	0,73	1,56	24,96	10,82	0,00	12,15	0,00	0,00	0,00	13,45	0,00	17,47
adult#06	0,00	2,01	15,81	13,90	0,00	15,14	0,00	0,00	0,00	14,51	0,00	16,14
adult#07	0,00	1,53	15,14	17,88	1,01	20,63	2,92	9,31	22,96	16,40	0,40	18,64
adult#08	0,50	6,68	34,61	13,40	0,00	15,81	0,00	5,58	25,79	14,18	0,00	18,14
adult#09	0,00	0,85	10,98	12,00	0,00	13,14	0,00	0,00	0,00	14,51	0,00	17,14
adult#10	0,00	2,14	22,63	1,98	0,00	4,99	0,00	0,00	0,00	12,17	0,00	37,60

4. RESULTS

In this section, the results obtained for the *in-silico* testing are presented. In order to illustrate the *in-silico* behavior, the curves for a particular subject are shown first. Figure 5 shows the glucose excursion, insulin infusion and IOB estimation for the subject Adult#05 from the UVA virtual patient cohort, comparing RL and manual strategies.

Table 1 shows *in-silico* results comparing the performances of the two strategies described in section 3. Total percentages of time in hypo- and hyperglycemia, as well as the maximum values of % across all simulation steps are shown. These results are only intented as a comparative between methods and not as a standarized metric.

5. DISCUSSION AND FUTURE WORK

Looking at Fig. 5, it can be clearly seen that the policy application successfully avoids hypoglycemia without increasing time in hyperglycemia when the patients sensitivity increases, while the manual scheme does not. For this case, it is also worth noting that insulin infusion is reduced when using the policy, showing that the RL strategy could also improve the insulin infusion profile.

When analyzing the results shown in table 1, it can be seen that the policy application outperforms the manual strategy for 9 of the 10 adult subjects, especially when considering time spent in hypoglycemia. Low percentages of overall time in hypoglycemia can be deceiving since the total period of simulation was 16 steps, but the “max value” column indicates whether that % of time was sustained during all the steps or not. When using the manual strategy, this column shows that all subjects, at least in one step, spent more than the accepted 4% of their time in hypoglycemia. These values are reduced to zero in 8 out of 10 subjects when using the policy generated via RL but for some subjects this leads to an increase of time in hyperglycemia. This issue seems to be a direct consequence of the reward design, since it focuses primarily in hypoglycemia. This could be overcome with a careful redesign of the reward and/or longer training time.

This particular RL work shows some similarities with the benchmark RL problem, such as the walking robot, in which a robot learns how to walk simply by knowing it is "wrong" to fall and "good" to stay up, and, as these examples show, with a simple reward design, the system should learn the correct policy. But at the same time, since it includes 10 different patients, it has an added layer of complexity: the policy is shared among all of them. This has the clear advantage of working as a general-use tool and the disadvantage of perhaps missing some key individual feature that cannot be compensated in the same way for every patient.

Taking this into consideration, future work includes personalized on-line policy training of the pre-trained general policy, with the reward redesigned so as to highly penalize "wrong" actions and ensure a better policy for all patients that were not well controlled by the general-use policy.

Thanks to the preliminary work in the area of Reinforcement Learning presented in this article, the working group will be addressing future investigations further examining the systems' dynamics role in the RL agent training, as well as the possibility of incorporating new states that represent these dynamics more accurately.

ACKNOWLEDGEMENTS

Research in this area is supported by the Argentine government (MINCYT/PICT2019-2554 and 2017-3211, UNLP/I253, CONICET/PIP2595) and the Comisión de Investigaciones Científicas (CICpBA).

REFERENCES

Battelino, T., Danne, T., Bergenstal, R.M., et al. (2019). Clinical Targets for Continuous Glucose Monitoring Data Interpretation: Recommendations From the International Consensus on Time in Range. *Diabetes Care*, 42(8), 1593–1603.

Carlucho, I., De Paula, M., and Acosta, G.G. (2019). Double Q-PID algorithm for mobile robot control. *Expert Systems with Applications*, 137, 292–307.

Colmegna, P., Garelli, F., De Battista, H., and Sánchez-Peña, R. (2018). Automatic regulatory control in type 1 diabetes without carbohydrate counting. *Control Engineering Practice*, 74, 22–32.

Dalla Man, C., Micheletto, F., Lv, D., et al. (2014). The UVA/PADOVA type 1 diabetes simulator: New features. *Journal of Diabetes Science and Technology*, 8(1), 26–34.

Daskalaki, E., Diem, P., and Mougiakakou, S.G. (2016). Model-free machine learning in biomedicine: Feasibility study in type 1 diabetes. *PLoS ONE*, 11(7), 1–20.

De Paula, M., Ávila, L.O., and Martínez, E.C. (2015). Controlling blood glucose variability under uncertainty using reinforcement learning and Gaussian processes. *Applied Soft Computing Journal*, 35, 310–332.

Fox, I. and Wiens, J. (2019). Reinforcement Learning for Blood Glucose Control: Challenges and Opportunities. *Workshop in the 36th International Conference on Machine Learning.*

Fushimi, E., Serafini, M.C., De Battista, H., and Garelli, F. (2020). Automatic glycemic regulation for the pediatric population based on switched control and time-varying IOB constraints: an in silico study. *Medical and Biological Engineering and Computing.*

Gottesman, O., Johansson, F., Komorowski, M., et al. (2019). Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1), 16–18.

Haidar, A. (2016). The Artificial Pancreas: How Closed-Loop Control Is Revolutionizing Diabetes. *IEEE Control Systems*, 36(5), 28–47.

Lanzola, G., Toffanin, C., Di Palma, F., et al. (2015). Designing an artificial pancreas architecture: the AP@home experience. *Medical and Biological Engineering and Computing.*

Messori, M., Kropff, J., Del Favero, S., et al. (2017). Individually Adaptive Artificial Pancreas in Subjects with Type 1 Diabetes: A One-Month Proof-of-Concept Trial in Free-Living Conditions. *Diabetes Technology & Therapeutics*, 19(10), 560–571. URL `http://www.liebertpub.com/doi/10.1089/dia.2016.0463`.

Ngo, P.D., Wei, S., Holubová, A., et al. (2018). Control of Blood Glucose for Type-1 Diabetes by Using Reinforcement Learning with Feedforward Algorithm. *Computational and Mathematical Methods in Medicine*, 2018.

Nian, R., Liu, J., and Huang, B. (2020). A review On reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139, 106886.

Oroojeni Mohammad Javad, M., Agboola, S.O., Jethwani, K., et al. (2019). A Reinforcement Learning–Based Method for Management of Type 1 Diabetes: Exploratory Study. *JMIR Diabetes*, 4(3), e12905.

Rosales, N., De Battista, H., and Garelli, F. (2022). Hypoglycemia prevention: PID-type controller adaptation for glucose rate limiting in Artificial Pancreas System. *Biomedical Signal Processing and Control*, 71, 103106.

Ruan, Y., Wilinska, M.E., Thabit, H., and Hovorka, R. (2017). Modeling Day-to-Day Variability of Glucose-Insulin Regulation Over 12-Week Home Use of Closed-Loop Insulin Delivery. *IEEE Transactions on Biomedical Engineering*, 64(6), 1412–1419.

Sánchez-Peña, R., Colmegna, P., Garelli, F., et al. (2018). Artificial Pancreas: Clinical Study in Latin America Without Premeal Insulin Boluses. *Journal of Diabetes Science and Technology*, 12(5), 914–925.

Sun, Q., Jankovic, M.V., Budzinski, J., et al. (2019). A Dual Mode Adaptive Basal-Bolus Advisor Based on Reinforcement Learning. *IEEE Journal of Biomedical and Health Informatics*, 23(6), 2633–2641.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement Learning: An Introduction.* MIT Press, 1 edition.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction (second edition).* MIT Press, 2 edition.

Szepesvári, C. (2010). Algorithms for Reinforcement Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1), 1–103.

Tejedor, M., Woldaregay, A.Z., and Godtliebsen, F. (2020). Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial Intelligence in Medicine*, 104(August 2019), 101836.

Toffanin, C., Visentin, R., Messori, M., et al. (2018). Toward a Run-to-Run Adaptive Artificial Pancreas: In Silico Results. *IEEE Transactions on Biomedical Engineering*, 65(3), 479–488.

Vos, T. and Lim, S.S. (2020). Global burden of 369 diseases and injuries in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019. *The Lancet*, 396(10258), 1204–1222.

Watkins, C.J.C.H. (1989). Learning from delayed rewards.