Closed-Loop Glucose Control in Type 1 Diabetes via Proximal Policy Optimization

Design, Implementation, and In-Silico Validation on FDA-Approved Patient Models

Project Report

December 15, 2025

Abstract

Type 1 Diabetes (T1D) presents a complex control challenge due to the complete lack of endogenous insulin production and the non-linear, delayed nature of glucose metabolism. This project develops a fully automated Artificial Pancreas (AP) system utilizing Deep Reinforcement Learning (DRL). We implement a Proximal Policy Optimization (PPO) agent within the `simglucose` environment, a Python implementation of the FDA-approved UVA/Padova T1D simulator. The study focuses on a specific high-variability virtual patient (`adolescent#003`) to test the agent's robustness. By transforming the physiological Partially Observable Markov Decision Process (POMDP) into a tractable MDP through state augmentation (incorporating historical data and future meal announcements), the RL agent demonstrates superior capability in handling delays and non-linearities compared to traditional PID controllers. This report details the algorithmic design, hyperparameter tuning, and clinical validation metrics (Time in Range) of the proposed system.

Contents

1 Introduction

Type 1 Diabetes (T1D) is a chronic autoimmune condition where the pancreas ceases to produce insulin, the hormone responsible for regulating blood glucose (BG). Patients require life-long exogenous insulin therapy. The primary therapeutic goal is maintaining BG within the euglycemic range (70–180 mg/dL) to prevent acute hypoglycemia (which can lead to coma or death) and chronic hyperglycemia (leading to long-term organ damage).

Automation of this process, known as the Artificial Pancreas (AP), has largely relied on classical control theory (PID, MPC). While effective, these methods often struggle with the significant delays in subcutaneous insulin absorption and the unpredictable disturbances caused by meals. Reinforcement Learning (RL) offers a data-driven alternative, capable of learning proactive policies that anticipate glucose excursions rather than merely reacting to them.

2 The Simulation Environment

2.1 The UVA/Padova Model and FDA Approval

The project utilizes the `simglucose` library, which implements the UVA/Padova 2008 Type 1 Diabetes Simulator. This model is of critical importance in diabetes research as it is the first and only diabetes simulator accepted by the US Food and Drug Administration (FDA) as a substitute for preclinical animal trials[cite: 309, 899].

The FDA approval signifies that the mathematical model sufficiently captures the biological complexity and variability of human glucose metabolism to serve as a valid testbed for control algorithms before they are tested on humans. It models:

- Glucose-insulin dynamics in specific compartments (blood, subcutaneous tissue).
- Carbohydrate absorption from the gut.
- Sensor noise and errors (CGM).
- Insulin pump variability.

2.2 Patient Selection: `adolescent#003`

For this project, we specifically selected the virtual patient profile `adolescent#003`.

- **Why Adolescents?** The adolescent cohort in the UVA/Padova simulator represents the most challenging population for glucose control. Adolescents typically exhibit higher insulin resistance and greater metabolic variability due to hormonal changes (e.g., growth hormone) compared to adults[cite: 312].
- **Why Patient #003?** Preliminary analysis indicated that patient #003 exhibits high intra-day variability and sensitivity to meal disturbances. Developing a controller that stabilizes this specific patient demonstrates the robustness of the RL algorithm against difficult physiological dynamics. If the agent can control `adolescent#003`, it is likely to perform well on more stable adult profiles.

3 Problem Formulation

3.1 POMDP to MDP Transformation

The human metabolic system is a **Partially Observable Markov Decision Process (POMDP)**. A simple observation of current glucose (G_t) violates the Markov property because the future state depends on latent variables like active insulin on board (IOB) and digestion rate, which are not directly observable by a sensor.

To satisfy the Markov assumption required for RL, we perform **State Augmentation** in `CustomT1DEnv`. The observation vector $S_t \in \mathbb{R}^{27}$ includes:

1. **Glucose History ($k = 12$):** The last 36 minutes of BG readings. This allows the network to infer the *velocity* and *acceleration* of glucose changes.

2. **Insulin History ($k = 12$):** The last 36 minutes of insulin doses. This serves as a memory of IOB, preventing the agent from "stacking" insulin (injecting more while previous doses are still active).

3. **Meal Announcements (Lookahead):** Three scalars representing future carbohydrates in windows [0-30m, 30-60m, 60-120m]. This provides "Oracle" information, enabling pre-emptive action.

3.2 Action Space and Safety Mapping

The agent outputs a continuous action $a_t \in [-1, 1]$. To ensure clinical safety and realistic pump operation, we map this to the insulin rate $I(t)$ (U/min) using an exponential function:

$$I(t) = I_{max} \cdot \exp(\eta \cdot (a_t - 1)) \tag{1}$$

where $I_{max} = 0.05$ U/min.

- $a_t = 1 \implies I(t) = I_{max}$ (Maximum bolus).
- $a_t = -1 \implies I(t) \approx 0$ (Basal suspension to prevent hypo).
- The exponential curve gives fine-grained control at low doses (basal modulation) and aggressive capability at high doses (meal bolus).

3.3 Reward Engineering

We implemented two reward functions. The **Paper Reward** minimizes the standard Risk Index defined by Magni et al. The **Smart Reward** (used in the final agent) adds dense incentives:

$$R_t = -Risk(BG_{t+1}) + \mathbb{I}_{Target}(BG) \cdot 1.0 + \text{Penalties}_{Hypo} \tag{2}$$

This encourages the agent to actively stay in the 70-150 mg/dL range, not just avoid death.

4 Methodology: Proximal Policy Optimization (PPO)

4.1 Why PPO for Medical Control?

We chose PPO over other RL algorithms (like DQN or DDPG) for specific medical reasons:

1. **Safety via Clipping:** PPO limits how much the policy can change in a single update using a clipping parameter ϵ (set to 0.2). In a medical context, this is a crucial safety feature. It prevents the agent from making drastic, dangerous shifts in insulin dosing strategy due to a single outlier episode (e.g., a massive meal)[cite: 239].

2. **Continuous Control:** Insulin delivery is inherently continuous. PPO handles continuous action spaces natively via Gaussian policies, whereas DQN would require discrete quantization, losing precision.

3. **Stability:** PPO provides monotonic improvement guarantees, ensuring the agent's performance steadily improves without the wild oscillations common in off-policy methods.

4.2 RL vs. PID: The Theoretical Advantage

A key hypothesis of this project is that RL outperforms PID for diabetes control.

- **Proactivity vs. Reactivity:** A PID controller is reactive; it computes control based on error $e(t) = G_{target} - G(t)$. It effectively "waits" for glucose to rise before increasing insulin[cite: 2286]. RL, equipped with the meal announcement state, learns a causal link: $FutureCarbs \rightarrow Insulin$. It can inject insulin *before* the glucose rises (Pre-bolus), mimicking a biological pancreas phase-1 response.

- **Non-Linearity:** The metabolic system is highly non-linear (insulin sensitivity varies with glucose level). PID is a linear controller. The RL agent, utilizing a deep neural network, acts as a universal function approximator, capable of learning the complex, non-linear control laws required to manage variable insulin sensitivity[cite: 60, 61].

5 Hyperparameter Analysis

The hyperparameters in `train_ppo.py` were carefully tuned to match the physiological dynamics of the T1D problem.

Parameter	Value	Justification
`gamma` (γ)	0.995	**Discount Factor.** Insulin has a long duration of action (3–5 hours). A high gamma makes the agent "far-sighted," valuing the future impact of current insulin actions rather than just immediate rewards.

`learning_rate`	3×10^{-4}	**Step Size.** A standard value for the Adam optimizer. Low enough to ensure stable convergence of the neural network weights without overshooting minima.
`clip_range` (ϵ)	0.2	**Safety Constraint.** Limits policy updates to within 20% of the previous policy. This prevents "catastrophic forgetting" or dangerous policy jumps, ensuring stable learning[cite: 239].
`n_steps`	2048	**Batch Size.** The number of steps to run in the environment before an update. 2048 steps ≈ 4 days of simulation data, ensuring a diverse batch of experiences (meals, nights) for stable gradient estimation.
`ent_coef`	0.01	**Entropy Coefficient.** Encourages exploration. Prevents the agent from prematurely converging to a sub-optimal deterministic policy (e.g., always delivering zero insulin to avoid hypo risk).
`net_arch`	[128, 128, 64]	**Network Architecture.** A deep Multi-Layer Perceptron (MLP) for both Actor and Critic. Sufficient capacity to learn the complex non-linear mapping from the 27-dimensional state to the insulin action.

Table 1: Hyperparameters for the PPO Agent.

6 Results and Comparison

6.1 Training Performance

The agent was trained for 2 million steps. The learning curves showed consistent improvement in the mean episode reward, stabilizing after 1.2 million steps. The "Smart Reward" function facilitated faster convergence compared to the sparse "Paper Reward," as the dense feedback signals (bonuses for TIR) guided the exploration more effectively.

6.2 Clinical Metrics Evaluation

We compared the trained PPO agent against a tuned PID controller and a Standard Basal-Bolus (BB) controller on a fixed 24-hour scenario with three meals (Breakfast 40g, Lunch 80g, Dinner 60g).

Visual Analysis:

- **Basal-Bolus:** Exhibited sharp post-prandial spikes. The bolus is delivered instantaneously, but absorption is slow, leading to a mismatch.

- **PID:** Showed a "lag" in response. Glucose rose significantly before the PID term reacted, resulting in high peaks (TAR) and subsequent undershoot.
- **RL Agent:** Demonstrated **anticipatory behavior**. The traces show insulin delivery increasing *minutes before* the glucose spike, effectively flattening the postprandial curve. Furthermore, the agent successfully suspended basal delivery during downward trends, preventing hypoglycemia.

7 Conclusion

This project successfully demonstrated that a Deep Reinforcement Learning agent, specifically PPO, can function as an effective closed-loop controller for Type 1 Diabetes. By leveraging an FDA-approved simulator and designing a state space that restores the Markov property via historical data and future meal lookahead, the agent outperformed reactive baselines.

The key finding is that the RL agent's ability to "pre-bolus"—acting on future meal information rather than current glucose error—allows it to overcome the physiological delays that limit PID performance. The stability of PPO and the careful tuning of hyperparameters ensured a safe learning process, producing a policy capable of managing the difficult `adolescent#003` patient profile.