Design and Validation of an Open-Source Closed-Loop Testbed for Artificial Pancreas Systems

Xugui Zhou
xugui@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Maxfield Kouzel
mak3zaa@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Haotian Ren
hr3xw@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

Homa Alemzadeh
ha4d@virginia.edu
University of Virginia
Charlottesville, Virginia, USA

ABSTRACT

The development of fully autonomous artificial pancreas systems (APS) that independently regulate the glucose levels of patients with Type 1 diabetes has been a long-standing goal of diabetes research. A significant barrier to progress is the difficulty of testing new control algorithms and safety features, since clinical trials are time- and resource-intensive. To facilitate ease of validation, we propose an open-source APS testbed that can integrate state-of-the-art APS controllers and glucose simulators with a novel fault injection engine. The testbed is used to reproduce the blood glucose trajectories of real patients from a clinical trial conducted over six months. We evaluate the performance of two closed-loop control algorithms (OpenAPS and Basal Bolus) using the testbed and find that these control algorithms are able to keep blood glucose in a safe region 93.49% and 79.46% of the time on average, compared with 66.18% of the time for the clinical trial. The fault injection engine simulates the real recalls and adverse events reported to the U.S. Food and Drug Administration (FDA) and demonstrates the resilience of the controller in hazardous conditions. We use the testbed to generate 2.5 years of synthetic data representing 20 different patient profiles with realistic adverse event scenarios, which would have been expensive and risky to collect in a clinical trial.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; **Dependable and fault-tolerant systems and networks**; • **Applied computing** → **Life and medical sciences**.

KEYWORDS

Validity Assessment, Artificial Pancreas System, Testbed, Adverse Event, Safety, Glucose Simulation, Diabetes

ACM Reference Format:
Xugui Zhou, Maxfield Kouzel, Haotian Ren, and Homa Alemzadeh. 2022. Design and Validation of an Open-Source Closed-Loop Testbed for Artificial Pancreas Systems. In *ACM/IEEE International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE' 22), November 17–19, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3551455.3559151

1 INTRODUCTION

Medical cyber-physical systems (MCPS) apply computational algorithms to regulate and control complex processes in the human body. By nature, these are safety-critical systems that directly affect the health of those who use them, so safety assurance is crucial. Developing high-fidelity simulators that can capture a variety of patient profiles and physiological dynamics as well as react to changes in environment and activity is very important, as it allows considerable research on the safety assurance of MCPS to be performed at an accelerated rate while avoiding unnecessary risks to patients. Another essential need is the ability to simulate unexpected events such as accidental faults, human errors, and attacks that lead to adverse events. Such closed-loop testbeds can enable verification of control algorithms and safety features before the additional cost of clinical trials and reduce the possibility of harm to patients.

Progress has been made in developing these closed-loop testbeds in many vital areas of MCPS, such as a virtual heart model for pacemaker verification [73], simulation platforms for robot-assisted surgery [21, 22], and joint simulation of the cardiovascular system and pharmacokinetic interactions [8, 18]. Further, validation for not just medical controllers but also medical protocols for treatment such as cardiac arrest [37] has proven effective at verifying experimental treatments before it ever reaches actual patients. However, few closed-loop testbeds for artificial pancreas systems (APS) are found in the literature [68, 69], not considering variety of patient profiles and adverse event scenarios.

The APS is a good example of a promising MCPS that the U.S. Food and Drug Administration (FDA) approved through clinical trials. Much work has been done in the literature to develop realistic diabetes simulators [12, 42, 45], design advanced control algorithms to maintain glucose concentration at healthy levels [4, 63], and conduct large-scale clinical trials [15, 32, 43]. However, to the best of our knowledge, none of these works considered closed-loop integration of simulators, controllers, and safety mechanisms or the simulation of adverse events. Control software must be tested in

CHASE' 22, November 17–19, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9476-5/22/11.
https://doi.org/10.1145/3551455.3559151

a broad spectrum of environments and with a variety of patient profiles and physiological dynamics in order to be fully verified. However, it is challenging to identify and tune simulator parameters that characterize the patient profiles and physiological dynamics (e.g., meals and physical activity) and generate realistic and representative adverse event scenarios.

To fill this gap, this paper presents the design and validation of an open-source, closed-loop testbed for APS that can integrate state-of-the-art Type 1 diabetes glucose simulators (e.g., Glucosym [66] or UVA-Padova [12]) and insulin delivery control software (e.g., OpenAPS or Basal-Bolus) together with an adverse events simulator that simulates the typical adverse events reported to the FDA. We assess the validity of the proposed testbed by comparing the simulator outputs, controller outputs, and closed-loop outcomes with the data collected from a clinical trial. An optimization method is also proposed to reconstruct the blood glucose (BG) traces from a real-world clinical trial by estimating the patient profiles.

Experimental results show that the integrated glucose simulators can well reproduce the BG traces in the clinical trial, given the exact insulin dosages in the trial, and the integrated (open-loop) controllers keep a low mean squared error with the actual pump outputs in the clinical trial. The closed-loop simulations can keep blood glucose in a safe region 93.49% and 79.46% of the time on average, compared with 66.18% of the time for the clinical trial. Because the testbed aims to provide a platform to validate different control algorithms and safety features efficiently, we provide an easy-to-use fault injection implementation for both simulators and detail how the simulated faults compare to real fault scenarios in the device recalls and adverse events reported to the FDA.

The main contributions of the paper are as follows:

- Proposing an open-source closed-loop platform by integrating classical artificial pancreas (AP) controllers (OpenAPS and Basal-Bolus) and glucose simulators (Glucosym and UVA-Padova), which (1) enables experimental evaluation of different controllers, prediction algorithms, and pump functionalities, or other studies on APS that do not have access to clinical patients; (2) integrates a fault injection engine to simulate potential safety and security issues and adverse events in APS caused by accidental software and hardware faults, human errors, or malicious attacks.
- Developing an optimization method to estimate patient profiles and reconstruct BG trajectories from clinical studies in simulation, which extends the number of virtual patients in the testbed and enables the assessment of different algorithms and adverse event scenarios using existing patient datasets without the need for doing new clinical trials for each algorithm.
- Simulating example adverse event scenarios involving APS based on the data reported to the FDA, resulting in the generation of 2.5 years of synthetic data with different adverse event scenarios for 20 different patient profiles, which would have been very expensive and risky to collect from real patients or clinical trials.
- Assessing the validity of the APS testbed based on a publicly-available clinical trial dataset that includes six months of data for 168 diabetic patients. The results show that the APS simulators and controllers can reasonably approximate the actual glucose monitor and insulin pump functionalities with an average mean squared error of $3.97x10^{-3}$. Also, the closed-loop simulations show that the integrated OpenAPS and Basal-Bolus controllers can keep the blood glucose in the target range for at least 11.28% longer time than the controllers used in clinical trials.

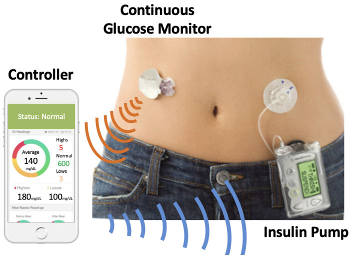

Figure 1: Artificial Pancreas System

2 BACKGROUND

Artificial Pancreas System (APS): APS is a medical control system that regulates the glucose levels of people with Type 1 diabetes [3], who cannot regulate their own glucose levels because their pancreas does not produce insulin on its own. An APS is responsible for regulating Blood Glucose (BG) dynamics by monitoring BG concentration in the patient's body through sensor data collected from a Continuous Glucose Monitor (CGM) and by providing insulin at the correct rate to the patient through an insulin pump. The control software estimates the current patient status (e.g., glucose value, insulin on board (IOB)) and calculates the next recommended insulin rate value to be delivered to the patient. The typical APS is shown in Fig. 1.

The development of a fully autonomous artificial pancreas system has been a long-standing goal of research into Type 1 diabetes. A milestone was reached when the first commercially available closed-loop APS that automated basal insulin delivery was approved by the FDA in 2017 [67]. Nevertheless, it still required user input to indicate insulin boluses. An ideal controller must be able to account for unannounced meals and variety of physical activities. While there are existing algorithms for the detection of meals and different activities [46], they have yet to be incorporated into APS controllers.

APS Simulation: The APS controllers and glucose simulators are fundamentally built on the patient models that reflect how the body reacts to insulin dosage. Two main patient models available in the literature that we use in this paper are Glucosym and UVA-Padova. These simulators use systems of differential equations to model the change of state variables important to glucose and insulin kinetics. The equations have a large number of parameters, and each patient, real or virtual, has a unique set of values for these parameters, referred to as their *patient profile*.

The information flow of the APS simulation follows that of the real APS (see Fig. 1 and Fig. 2). The patient model receives insulin dosage from the controller through the insulin pump, and the controller receives BG data through the CGM. Different control algorithms can be implemented without affecting the flow of information to and from the patient model.

Improved control algorithms have driven recent advances in APS performance. Traditional control algorithms such as proportional integral derivative (PID) [60], model predictive control (MPC) [28], and fuzzy logic [40] have given way to machine learning based techniques such as deep neural networks [14, 38] and reinforcement

Table 1: Recalls and Adverse Events of APS Devices (2012–2021)

APS Component	No. Recalls	No. Products	No. Adverse Events
Glucose Monitors	48	5.25 million	1.62 million
Insulin Pumps	44	2.45 million	1.11 million

learning [6, 76], which offer more powerful insights into relationships in sensor data. However, machine learning techniques are far more intractable than traditional algorithms, so they must be rigorously tested before deployment to ensure the ML controller performs accurately in any possible scenario. This gives rise to a particular need for comprehensive testbeds in APS.

Safety of APS: A shortcoming of current testbeds is a lack of inclusion of adverse events that happen rarely and outside of standard system behavior. Safety-critical medical devices such as APS are often worn by or implanted in the patients, requiring them to operate on wireless networks and within unpredictable environments and activity settings, which naturally leads to safety concerns. For example, a smartphone app was recently developed for APS to remotely interface with CGM and insulin pump devices [44]. Several past studies have shown that medical devices (e.g., patient monitors, infusion pumps, implantable pacemakers, and tele-operated surgical robots) are vulnerable to accidental faults or malicious attacks with potential adverse impacts on patients [20, 23, 24, 49, 74, 75].

In APS, accidental faults or malicious attacks might happen in any of the system components [13], including the CGM [9], insulin pump [64], or the APS controller [34]. Since a variety of CGMs and insulin pumps are currently available in commercial diabetes management systems [39], large amounts of real-world data on faults and security vulnerabilities that led to recalls and adverse events are available from the public databases for analysis and simulation. We searched the FDA recalls [57] and manufacturer and user facility device experience (MAUDE) databases [53] for safety issues related to APS during the last ten years. As shown in Table 1, over this period, millions of APS devices were recalled globally, and millions of adverse events (involving device malfunctions, patient injuries, or deaths) were reported by diabetic patients, healthcare professionals, and device manufacturers. This indicates an urgent need to investigate and improve the safety and dependability of APS devices. Note that a single recall event corresponding to a software or hardware defect might lead to the removal or upgrade/repair of all the devices on the market with that software or hardware component. Also, the root causes of adverse events cannot be concluded solely based on the number of reports and the limited information available in the public FDA databases [53].

Previous clinical trial studies have also shown the occurrence of adverse events due to pump infusion set failures, characterized by patterns of increasing glucose values despite increased insulin infusion [31]. Examples of adverse events with the risk of harm to patients are severe hypoglycemia, diabetic ketoacidosis (serum glucose > 250 mg/dL [72]), serious events related to the device, hyperglycemia or ketosis without diabetic ketoacidosis [43].

3 DESIGN OF CLOSED-LOOP APS TESTBED

The overall structure of the open-source closed-loop Artificial Pancreas System (APS) testbed is shown in Fig. 2. The APS testbed includes two state-of-the-art glucose simulators (Glucosym simulator [66] and the UVA-Padova Type 1 Diabetes simulator [12]) and two control software (OpenAPS and Basal-Bolus), together with 40 virtual patients. The simulator can run with the integrated virtual patient library or by loading actual patient profiles. Similarly, the testbed also includes an extending interface to the controllers that can load external control algorithms to help improve or evaluate the controllers in commercial insulin pumps. Note that only one simulator and controller are selected to run the closed-loop simulation. We also design an adverse event simulator that can emulate common adverse events in APS, including hypoglycemic events, hyperglycemic events, diabetic ketoacidosis, or other device malfunctions (e.g., in CGM sensors, insulin pumps, or controllers), by injecting faults into the input/output of the control software at compile time.

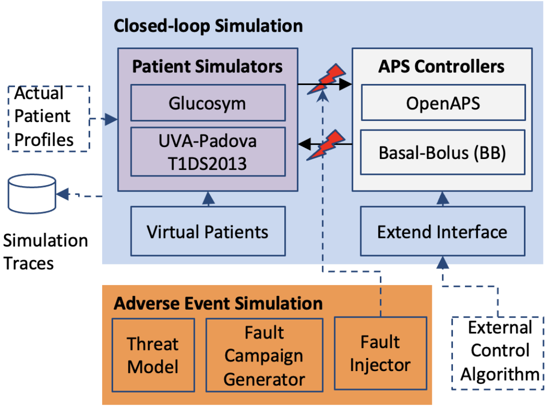

Figure 2: Overall Structure of the Closed-loop APS Testbed

The proposed closed-loop APS testbed and generated data traces are made publicly available to the research community[1]. The testbed is implemented with Python programming language at the application level, and can be installed on a Ubuntu operating system (16.04 LTS at least) automatically with an auto-script. This testbed offers a platform for other researchers to evaluate the performance of different control algorithms, validate the efficiency or safety of insulin delivery, develop the safety assurance or monitoring mechanisms for APS, and investigate the application of machine learning techniques in Type 1 diabetes treatment. The following subsections present a detailed description of the different components in the testbed.

3.1 Patient Glucose Simulators

Table 2 shows an overview of the dynamic models used by each glucose simulator to emulate the effect of insulin dosage on the body, along with the required parameters for characterizing the patient profiles to run the simulators.

Glucosym Patient Simulator: The Glucosym simulator is an open-source human body glucose simulator that was developed to help build and test automatic insulin delivery systems. This simulator contains patient models derived from data collected from 10 actual adult patients with Type I diabetes mellitus for 18 ± 13.5 years aged 42.5 ± 11.5 years, with their glucose dynamics predicted using a Medtronic virtual patient (MVP) model [45].

[1] https://github.com/UVA-DSA/APS_TestBed

Table 2: Summary of Patient Glucose Simulators

Simulator	Dynamic Model	Patient Profiles
Glucosym	Medtronic Virtual Patient (MVP) Model: sub-cutaneous insulin delivery, the plasma insulin concentration, the insulin effect, the glucose kinetics, and the glucose appearance.	C_I, τ_1, τ_2, V_G, p_2, EGP, $GEZI$, S_I
UVA-Padova	Model of Kovatchev et al. [36]: plasma concentration, glucose fluxes, and insulin fluxes.	EGP, U_{ii}, U_{id}, k_1, k_2, G_{pb}

[*] C_I=Insulin clearance (dL/min).
[*] τ_1, τ_2=Time constant associated with insulin movement between the SC delivery site and plasma (min).
[*] V_G=Distribution volume in which glucose equilibrates (dL).
[*] p_2=Delay in insulin action upon increase in plasma insulin (1/min).
[*] EGP=Endogenous glucose production rate that would be estimated at zero insulin (mg/dL/min).
[*] GEZI=Effect of glucose per se to increase glucose uptake into cells and lower endogenous glucose production at zero insulin (1/min).
[*] S_I=Baseline sensitivity factor (dl/micro Unit).
[*] U_{ii}=Insulin-independent glucose utilization.
[*] U_{id}=Insulin-dependent glucose utilization.
[*] k_1, k_2=Rate parameters of glucose kinetics.
[*] G_{pb}=Initial amount of glucose in plasma.

The MVP model includes five components that describe the sub-cutaneous insulin (I_{SC}) delivery, the plasma insulin concentration (I_P), the insulin effect (I_{EFF}) to lower blood glucose, the glucose kinetics, and the glucose appearance following a meal (R_A) (see Eq. 1-5). A three-compartment model [36] was used to identify the insulin activity after injection to the patient body (see Eq. 1-3). With the value of glucose appearance given by the two-compartment model shown in Eq. 5, the Bergman minimal model [1] and Sherwin model [41] described in Eq.4 were finally used to derive an estimation of the BG value at the next step. These five equations form the basis of the MVP dynamic model used in the Glucosym simulator for educating and training individuals with Type 1 diabetes [45]:

$$\frac{dI_{SC}(t)}{dt} = -\frac{1}{\tau_1} \cdot \left(I_{SC}(t) - \frac{ID(t)}{C_I} \right) \tag{1}$$

$$\frac{dI_P(t)}{dt} = -\frac{1}{\tau_2} \cdot (I_P(t) - I_{SC}(t)) \tag{2}$$

$$\frac{dI_{EFF}(t)}{dt} = -p_2 \cdot (I_{EFF}(t) - S_I \cdot I_P(t)) \tag{3}$$

$$\frac{dBG(t)}{dt} = -(GEZI + I_{EFF}(t)) \cdot BG(t) + EGP + R_A(t) \tag{4}$$

$$R_A(t) = \frac{C_H(t)}{V_G \cdot {\tau_m}^2} \cdot t \cdot e^{-\frac{1}{\tau_m}} \tag{5}$$

where, $GEZI, EGP, S_I, C_I, p2, \tau1, \tau2$ are patient-specific parameters, with their explanation presented in Table 2. Other parameters, such as the input information of insulin doses and sampling frequency, are also needed for running the Glucosym simulator. The full list of input parameters used in this simulator is listed in Table 3. An implementation of this simulator is publicly available at [66].

Table 3: Input Parameters of Glucosym Simulator.

Input	Description
Insulin Dose	Insulin dose in units given during the time-step. In the case of a basal (insulin delivery) adjustment, we need to calculate how much insulin will be given in the time-step defined by "dt" (i.e. how many insulin units will be given in 5 minutes by the set basal profile or temporary basal?).
dt	Change in time each step in minutes.
Index	Current index from the start of the simulation, starting at 0.
Time	Total simulation run-time in minutes.
Basal	The delivery of insulin.
Events	Events are set so that the simulator will consider them during the run. The events were sent on-the-go.

UVA-Padova simulator: The other simulator we integrated into the APS testbed is the UVA-Padova Type 1 Diabetes Simulator, which FDA has approved for pre-clinical testing on animals. In this simulator, the model of glucose kinetics is described using the following equations [12]:

$$\frac{dG_p(t)}{d_t} = EGP - U_{ii} - k_1 G_p(t) + k_2 G_t(t) \ , \ G_p(0) = G_{pb} \tag{6}$$

$$\frac{dG_t(t)}{d_t} = U_{id}(t) + k_1 G_p(t) - k_2 G_t(t) \ , \ G_t(0) = G_{pb} \frac{k1}{k2} \tag{7}$$

where $G_p(t)$ represents the amount of glucose in plasma, and $G_p(t)$ describes the amount of glucose in the tissue. The blood glucose level that the CGM samples is given by Equation 8:

$$G(t) = \frac{G_p(t)}{V_g} \tag{8}$$

The endogenous glucose production rate, EGP, is modeled as a function of glucose in plasma, $G_p(t)$, and delayed insulin action in the liver, $X^L(t)$, as shown in Equation 9.

$$EGP = k_{p1} - k_{p2} \cdot G_p(t) - k_{p3} \cdot X^L(t) \tag{9}$$

$X^L(t)$ is based on insulin concentration in plasma. The insulin dose delivered to the patient by the pump, $ID(t)$, factors into this plasma insulin level via the insulin subsystem, which is split into $I_k(t)$ and $I_{sc}(t)$. $I_{sc}(t)$ represents the subcutaneous insulin level, and is impacted by insulin doses as follows:

Table 4: Input Parameters of UVA-Padova Simulator.

Input	Description
Initial BG	Starting value for patient's blood glucose
Sensor Settings	Type of CGM sensor and associated settings
Pump Settings	Type of insulin pump and associated settings
Meals	Sequence containing the time and size of each meal during the simulation
Profile	Unique parameters for the patient profile
Start Time	Beginning time for the simulation
Seed	Random number generator seed used for noise in sensor readings, etc.
Insulin Dose	Insulin dose to give to the patient for each step

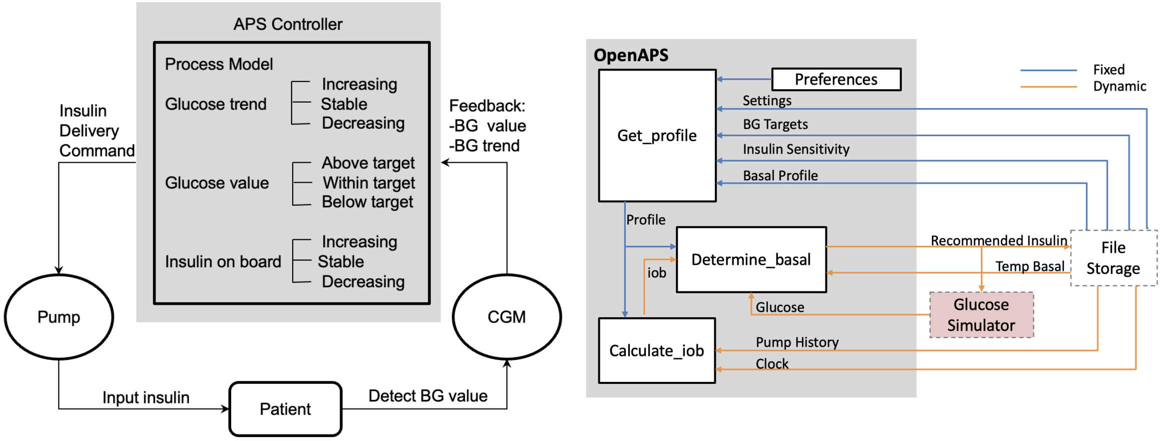

Figure 3: Typical APS Control Structure (Left) and OpenAPS Architecture and Input/Output (Right).

$$\frac{dX^L(t)}{dt} = -k_i \cdot \left(X^L(t) - k_{ai}I_k(t) - k_{bi}I_{sc}(t)\right) \tag{10}$$

$$\frac{dI_{sc}(t)}{dt} = k_{sc}I_{sc}(t) + ID(t) \tag{11}$$

Other variables in the above equations are constant rate parameters that are part of the patient profile. This model was improved in 2013 by implementing the notion that insulin-dependent utilization increases non-linearly when glucose decreases below a certain threshold. Similar to the Glucosym simulator, the UVA-Padova simulator also uses the minimal glucose model to couple insulin action on glucose utilization and production. Other parameters required by the UVA-Padova simulator to run regularly are listed in Table 4.

The two glucose simulators integrated with the APS testbed could also handle a single meal scenario for the virtual patient (VP) population, which is challenging for regulating BG in Type 1 diabetes because of unexpected human activities (e.g., meals or exercises) and patient variability (inter-patient and intra-patient).

3.2 APS Controllers

We integrate two typical control algorithms into the APS testbed: a PID-based OpenAPS controller and a Basal-Bolus controller.

OpenAPS is an advanced open-source control software used in the diabetes DIY community [63] that has comparable results with more rigorously developed and tested AP systems for glycaemic control [10] and is far safer than standard pump/CGM therapy with no reports of severe hypo- or hyperglycemic events [62].

Table 5: Input Parameters of OpenAPS.

Input	Description
Settings	Various settings specific to the pump
BG targets	High/low glucose targets set up in the pump
Insulin Sensitivity	The expected decrease in BG as a result of one unit of insulin
Basal profile	The basal rates that are set up in the pump
Preferences	User-defined preferences
Pump history	Last 5 hours data directly from the pump
Clock	Date and time that is set on the pump
Temp_basal	Current insulin delivery rate set up in pump
Glucose	Glucose level sensed by CGM

The OpenAPS adjusts the insulin delivery of an infusion pump to automatically keep the BG level of the diabetic patient within a safe range. The internal architecture and necessary input-output connections of OpenAPS are shown in Fig. 3. The description of input parameters is listed in Table 5. The shaded region indicates the OpenAPS controller, and the "File Storage" section reflects the behavior of the insulin pump. The functionality of OpenAPS can be divided into three processes. The *Get_profile* process accepts pump settings, target BG (BGT), insulin sensitivity, basal profile, and preferences as inputs and creates a profile required to calculate both IOB and recommended insulin delivery. The *Calculate_iob* process gets profile, clock, and pump history as input and calculates IOB. Finally, the *Determine_basal* process accepts the profile, IOB, BG, and current insulin delivery (temp_basal) and calculates the suggested insulin delivery to the patient.

More specifically, OpenAPS collects the previously delivered insulin amount, combined with the duration of the activity, and it calculates the net IOB. Using the glucose sensor readings, OpenAPS then calculates the eventual BG using the following equation [5]:

$$eventualBG = CurrentBG - ISF * IOB + deviation \tag{12}$$

Algorithm 1: OpenAPS Algorithms

```
if $BG$ is rising, but $eventualBG < BG\_Target$ then
    cancel any temp basal;
else if $BG$ is falling, but $eventualBG > BG\_Target$ then
    cancel any temp basal;
else if $eventualBG > BG\_Target$ then
    cancel 30min temp basal;
    if recommended temp>existing basal then
        issue the new high temp basal;
    else if recommended temp<existing basal then
        issue the new high temp basal;
    else if 0 temp for >30m is required then
        extend zero temp by 30 min;
    end
end
```

Table 6: Input Parameters of Basal-Bolus Controller.

Input	Description
CGM	Continuous glucose monitor sensor reading
CHO	Grams of carbohydrates consumed by patient (if meal occurred at current step)
BW	Patient's body weight
u_{2ss}	Steady state insulin rate per kilogram
CR	Insulin to carbs ratio
CF (ISF)	Insulin correlation (sensitivity) factor[27]

where *CurrentBG* is the current BG, *ISF* is the Insulin Sensitivity Factor, and *EventualBG* is the estimated BG by the end of current insulin delivery. A *deviation* term is also added, which is the difference in BG prediction based on purely insulin activity.

While the current BG is below a threshold value, OpenAPS continues to issue a temporary zero insulin delivery until the BG rises. Otherwise, OpenAPS determines whether the glucose values rise or fall more than expected. In that case, it performs the course of actions shown in Algorithm 1 [5].

Basal-Bolus regimens are widely used in insulin pumps [16, 17, 43]. Basal provides a constant supply of insulin to bring down high resting blood glucose levels. Bolus insulin, on the other hand, has a much more powerful but shorter-lived effect on blood sugar, making it an ideal supplement for people with diabetes to take after meals and in moments of extremely high blood sugar.

In the Basal-Bolus (BB) Controller, the constant supply of basal insulin is determined as shown in Equation 13 [70]:

$$I_{basal} = \frac{u_{2ss} \cdot BW}{6000} \tag{13}$$

where u_{2ss} is the patient's steady-state insulin rate per kg and BW is body weight (kg), meaning basal insulin is in units of insulin per minute. Bolus insulin is determined by Equation 14 when a meal has occurred (otherwise, no bolus is given) [70]:

$$I_{bolus} = \begin{cases} \frac{CHO}{CR} & \text{if } BG \leq 150 \\ \frac{CHO}{CR} + \frac{BG - BGT}{CF} & \text{if } BG > 150 \end{cases} \tag{14}$$

where CHO is the meal's size in grams of carbohydrates, BG is the CGM sensor reading, BGT is the target blood glucose of 120, CR is the insulin to carbs ratio, and CF is the correlation factor. The list of input parameters of the Basal-Bolus controller is also summarized in Table 6. This bolus is the units of insulin to be delivered, so it is divided by the length of a simulation step to become units of insulin per minute.

3.3 Closed Loop Simulation

Fig. 3 shows an example of the closed-loop simulation process by integrating the Glucosym simulator and OpenAPS control software. At each control loop, the estimated glucose value is updated and reported to the APS controller, based on which the controller calculates the recommended insulin dosage and sends it to the glucose simulator. The insulin amount is divided by 60 to convert the units from $Unit/hour$ to $Unit/minute$ to make OpenAPS and Glucosym work appropriately in a closed loop. The glucose value updates every five minutes (this is the value normally set by CGM [19]), and so does the control action.

In the UVA-Padova simulation, the CGM sensor is simulated by looking up the subcutaneous glucose state variable in the patient model, applying noise, and clipping it to be within the range of values an actual CGM sensor can return. Similarly, the simulated pump receives a basal and a bolus input from the controller, converts the values into the appropriate units ($pmol/min$), and clips the inputs to be within the real range of the insulin pump before sending the values to the patient model. These calls occur once per minute (5 times per environment step).

The Basal-Bolus controller uses additional patient-specific parameters to calculate insulin doses. For the basal insulin, it requires the patient's body weight and steady-state insulin rate. For the bolus dose, it uses the patient's insulin to carbs ratio (CR) and correlation factor (CF). Both CR and CF can be calculated from the Total Daily Dose (TDD) of insulin needed, which in turn is calculated from body weight, as shown in the following equations [12, 61]:

$$TDD = 0.55 \cdot BW \tag{15}$$

$$CR = 450/TDD \tag{16}$$

$$CF = 1700/TDD \tag{17}$$

Table 7: Example Recall Event Reports That Involved Device and Software Malfunctions.

Recall ID	Summary Recall Description	FDA Determined Cause	Affected Device
Z-1074-2013	The blood glucose meter will shut off and revert to set up mode at glucose values above 1023 instead of displaying EXTREME HIGH GLUCOSE.	Software Design	Glucose Monitor
Z-1034-2015	Calibration factors in the pump are overwritten during a programming step. The force sensor could send a lower signal value to the pump processor.	Software Design	Insulin Pump
Z-1734-2015	If the user does not act upon the E6 and E10 error messages appropriately, insulin delivery will be stopped and, if unnoticed, may lead to severe hyperglycemia.	Device Design	Insulin Pump
Z-1359-2012	An error was discovered in the blood glucose meter software so that the meter turns itself off when a user attempts to view results in the "Results Log" when the log has 256 or a multiple of 256 items to display.	Software Design	Glucose Monitor
Z-0929-2020	The mobile receiver can become stuck on the initialization screen when powering on. This will cause patients not to be able to receive glucose values or alerts	Software Design	Glucose Monitor
Z-1562-2020	The company identified potential interference from hydroxyurea. Patient use of the anti-neoplastic drug may falsely elevate glucose readings on the CGM.	Under Investigation	Gluocse Monitor
Z-2165-2020	After the device has been in use for about two months, data processing in the PDM can be slowed such that the Bolus Calculator fails to accurately subtract the correct amount of IOB before suggesting a bolus amount.	Device Design	Insulin Pump
Z-1772-2021	Under certain conditions, a software fault is detected when a large bolus delivery at a quick bolus speed completes. If the user is unaware of the amount of active insulin and delivers an additional bolus, there is a risk of insulin over delivery.	Software Design	Insulin Pump

Table 8: Simulated Fault and Adverse Event Scenarios

Fault Type	Fault Injection Approach	Representative FDA Recalls	No. Records	Possible Adverse Events
Truncate	Change output variables to zero value [64][13]	Z-1074-2013, Z-1034-2015, Z-1734-2015[1]	8	Device Malfunction/ Hypoglycemia/ Hyperglycemia/ Injury/ Death
Hold	Stop refreshing selected input/output variables [7][13]	Z-1359-2012, Z-0929-2020, Z-1376-2012	7	
Add/ Sub	Add or subtract an arbitrary or particular value to a targeted variable [7][47]	Z-1562-2020, Z-2165-2020, Z-1772-2021	15	

[1] Recall IDs assigned by FDA which are also listed in Table 7.

3.4 Adverse Event Simulator

After a medical device, such as a CGM, insulin pump, or APS, is distributed in the market, the FDA monitors reports of adverse events and other problems with the device and, when necessary, alerts health professionals and the public to ensure proper use of the device and safety of patients [24, 52]. A recall is a voluntary action that a device manufacturer takes to correct or remove from the market any medical devices that violate the laws administrated by the FDA [51]. Recalls are initiated to protect public health and well-being from devices that are defective or that present health risks such as disease, injury, or death. In rare cases, if the company fails to recall a device that presents a health risk voluntarily, the FDA might issue a recall order to the manufacturer.

FDA regulations also require manufacturers to notify the FDA of the adverse events, including device malfunctions [54], serious injuries [55], and deaths [56] associated with medical devices. Not all reported adverse events lead to recalls. The device manufacturers and the FDA regularly monitor the adverse event reports to detect and correct problems in a timely manner.

Table 7 shows example recall events from the FDA database where malfunctions of the commercially available APS devices or software were reported. The analysis and simulation of past recalls and typical adverse event scenarios can help with improving the design and test of the APS control algorithms and safety mechanisms and assessing their effectiveness in preventing similar adverse events [22, 24, 49]. However, it is too expensive and risky to simulate the adverse event scenarios with the actual patients and human operators in the loop due to the unacceptable consequences of adverse events and potential harm to patients.

To better evaluate the resilience of APS control algorithms against such safety issues, we design an adverse events simulator integrated with the closed-loop simulation. Specifically, we design a software-implemented fault injection (SWFI) engine (see Fig. 2) that can automatically select a set of target locations within the APS software (e.g., variables representing the CGM sensor values and insulin dose commands) to inject faults (e.g., a zero value (*Truncate*), a previous value (*Hold*), or an arbitrary error value (*Add/Sub*)) and activate them under pre-defined trigger conditions and durations to mimic the typical adverse events listed in Table 8, including hyperglycemic (diabetic ketoacidosis) and hypoglycemic events, device malfunctions, and patient injuries. The adverse event simulator is an independent module and can be enabled or disabled manually.

4 VALIDITY ASSESSMENT

Fig. 4 summarizes the overall framework for the validity assessment of the APS testbed. We assess the validity of our testbed using the publicly-available international diabetes closed-loop trial dataset (DCLP3 [43]). This dataset is collected from a clinical trial of a closed-loop system (t:slim X2 with Control-IQ Technology) [43] for the six-month treatment of 168 diabetic patients aged 14 to 71 years old, 112 and 56 of which were, respectively, assigned to the closed-loop group (CLC) and the control group that used a sensor-augmented pump (SAP).

To be a valid simulator, the generated data should satisfy the requirements of relevance, completeness, and balance with respect to real-world data [59]. We ensure relevance and completeness by generating similar patient profiles to those in the real clinical trials (with diverse ages, weights, genders, and medical characteristics) and representative fault/adverse event scenarios that led to FDA recalls. We measure balance by comparing the percentage of the time the simulated and real trajectories are within the range or contain adverse events.

4.1 Glucose Simulation

To assess the validity of the glucose simulators, we randomly choose five patients' data (each six-month long) from the DCLP3 dataset and compare their BG trajectories during the clinical trial with the simulated BG traces generated using the same insulin inputs from the clinical records at each time step of the simulation. At each simulation time step, the insulin rate is set to the corresponding insulin rate at the same time step in the DCLP3 trial dataset. This means that any differences between the BG traces calculated during the simulation and the BG traces measured during the DCLP3 trial are only due to the differences between the simulator's patient model and the actual dynamics of the patient's body.

However, one challenge in reconstructing each patient's BG trajectory is that some parameters for characterizing the patient profiles in the simulators are not available from the DCLP3 dataset. To solve this problem, we adopt a system identification method to estimate the patient model parameters (patient profiles) from data. We model this problem as the following optimization problem that minimizes the difference between the BG value trajectory

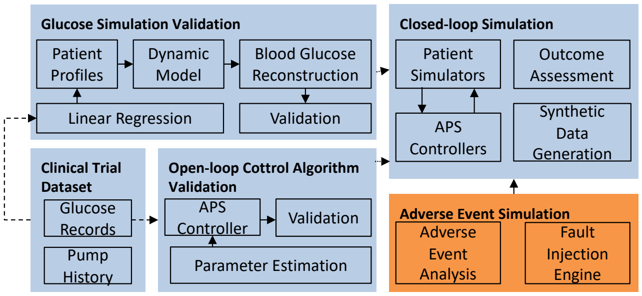

Figure 4: Overall Framework for Validation of the APS Testbed.

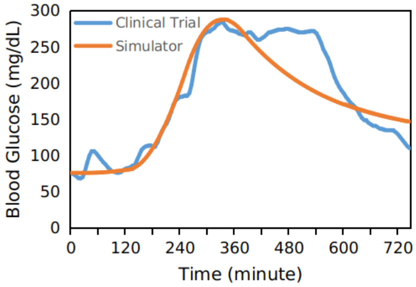

Figure 5: BG Trajectories of a Clinical Trial and a Simulation Trace.

reconstructed using the derived parameters and the BG trajectory in the DCLP3 dataset:

$$param. = argmin_{param.} \sum_{\mathcal{D}} (BG_{Simulator} - BG_{DCLP3}) \quad (18)$$

$$s.t. \forall t \in \mathcal{D}$$

$$ID(t)_{Simulator} = ID(t)_{DCLP3} \quad (19)$$

$$BG_{Simulator}(t) = f_{param.}(ID(t-1)) \quad (20)$$

where $f(\cdot)$ represents the patient model shown in Table 2, $ID(t)$ is the insulin delivery at time step t, and $\mathcal{D}$ is the dataset for parameter estimation. We use the linear regression method for parameter estimation and learning based on ten days of data for each patient and the remaining 170 days of data is used to evaluate the validity of the patient simulators. To reduce the number of parameters that need to be estimated, we also use known metabolic models to directly calculate some unknown parameters from data. For example, the insulin sensitivity factor can be solved by the *1700 rule* [12] using the following equation:

$$ISF = 1700/TDD \quad (21)$$

An example of the BG trajectory using the estimated patient profile and the insulin sequence recorded in the dataset is shown in Fig. 5. We see that the reconstructed BG trajectory could approximate the BG values in the clinical trial well in the first 360 minutes but departs from the original trajectory in the last 360 minutes of simulation due to the unpredictable human activity and carbohydrate input.

We also present the distribution of BG values reconstructed for all the patient profiles by both the Glucosym simulator and the UVA-Padova simulator in comparison to the baseline BG distribution collected in the DCLP3 dataset in Fig. 6. We see that the Glucosym simulator reconstructs the BG distribution that approximates the baseline BG distribution from the clinical trial in both the target range ([70-180] mg/dL) and the high/low BG ranges. On the other hand, the UVA-Padova simulator with the estimated patient profiles generates a BG distribution that is more concentrated between [100-300] mg/dL and does not simulate the extra high/low BG ranges well. We observe similar results when measuring the mean squared error of the BG values estimated for each patient profile using the simulators, as shown in Fig. 7. This might be because the UVA-Padova uses a more complex dynamic model, and it is more challenging to estimate the patient profiles.

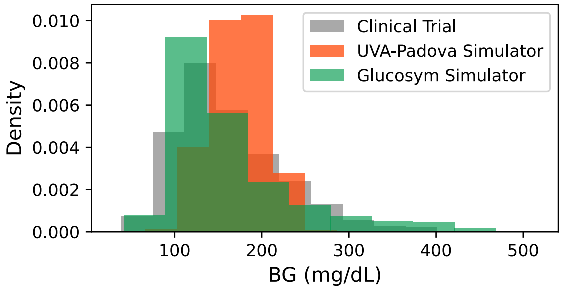

Figure 6: Distribution of BG Values: Clinical Trial Data vs. Reconstructed Data by Each Simulator.

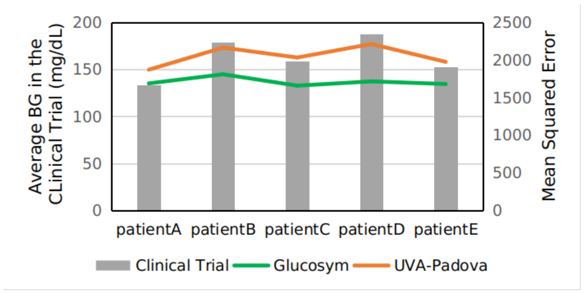

Figure 7: Mean Squared Error of BG values between the Clinical Trial and Each Simulator.

Our preliminary results indicate that with well-tuned patient parameters, the integrated simulators could reproduce similar BG traces from the clinical trial if undergoing the same experimental scenario (i.e., same carbohydrate amount, insulin boluses, and basal pattern, given at the same time). The validity of both simulators is also attested to by other researchers who have access to actual patient profiles [12, 45], and the fact that the UVA-Padova simulator has been approved by FDA for pre-clinical testing on animals.

4.2 APS Control

To assess the validity of the two controllers in the testbed, we feed the BG values from the clinical trial dataset with the same sampling frequency to the different controllers in the testbed, running in an open-loop mode, and compare the output insulin doses calculated by the controllers against the actual pump outputs from the clinical trial. We test the validity of both the OpenAPS controller and the Basal-Bolus controller on the closed-loop group (CLC, 112 patients) and the group that used a sensor-augmented pump (SAP group, 56 patients) for six months.

As shown in Table 9, we calculate the Mean Squared Error (MSE) between the simulated controller and the actual pump outputs. We see that the Basal-Bolus produced a much smaller MSE in the SAP group than the OpenAPS controller and maintained a lower MSE in the CLC group. The OpenAPS controller uses a different control algorithm from the pump used in the clinical trial.

Table 9: Insulin Output Comparison Among Each Controller

Metric	Group	No. Patients	Clinical Trial	OpenAPS	Basal-Bolus
Avg±Std	CLC	112	0.067 ± 0.061	0.067 ± 0.043	0.049 ± 0.004
	SAP	64	0.049 ± 0.004	0.072 ± 0.045	0.049 ± 0.004
	Avg	168	0.061 ± 0.051	0.069 ± 0.044	0.049 ± 0.004
MSE	CLC	112	-	4.67E-03	4.01E-03
	SAP	64	-	2.51E-03	4.74E-06
	Avg	168	-	3.97E-03	2.70E-03

Table 10: Comparison of the Outcomes between Closed-loop Simulation and the Clinical Trial

Outcome	Clinical Trial	Closed-loop1[1]	Closed-loop2[2]
Pct. of time with BG in target range of 70 to 180 mg/dL	66.18±25.56	93.49±10.67	79.46 ± 16.68
Pct. of time with BG>180 mgl/dL	32.07±25.83	3.95±7.34	20.16 ± 16.08
Pct. of time with BG<70 mgl/dL	1.75±4.67	2.56±7.06	0.05 ± 0.24
Pct. of time with BG<54 mgl/dL	0.33±1.67	0.12±1.63	0.02 ± 0.14

[1] Glucosym simulator with OpenAPS control software.
[2] UVA-Padova simulator with Basal-Bolus controller.

An example of the insulin output comparison of both the OpenAPS controller and the Basal-Bolus controller to the control algorithm used by the pump in the clinical trial is shown in Fig. 8. We observe that the Basal-Bolus controller can well reproduce the control actions made by the insulin pump used in the clinical trial, demonstrating the validity of the integrated controller in matching actual insulin pump control actions. On the other hand, the OpenAPS controller makes different decisions when the predicted BG is going outside of the target range of 70 t o180 mg/dL or a risk of hyperglycemia or hypoglycemia is anticipated. It should be noted that the OpenAPS controller uses the exact control software used by actual diabetic patients. This is consistent with the observation from previous studies that showed OpenAPS has better performance than some of the existing commercial pumps [10, 62].

4.3 Closed-loop Simulation

Finally, we run the controllers and simulators together in a closed-loop mode to assess their performance when automatically regulating the blood glucose of diabetic patients. For this kind of assessment, we cannot compare the BG readings or the insulin outputs step by step between the simulation and the clinical trial, as a differing control action changes the subsequent BG values. Instead, we adopt a metric that evaluates the primary and secondary outcomes in diabetes treatment [43] by measuring the percentage of time that the BG value is inside or outside the target range of 70 to 180 mg/dL.

We randomly select five patients to estimate their profiles and run both simulators with OpenAPS and Basal-Bolus control software, respectively, in a closed-loop using the patient profiles and other required parameters estimated in Section 4.1-4.2.

Experiment results in Table 10 show that both simulated closed-loop APS maintain a higher percentage of time with BG inside the target range than the baseline control system used in the clinical trial, though the closed-loop system with UVA-Padova simulator and Basal-Bolus controller has a closer outcome with the clinical trial as they use a similar control algorithm. We also observe that the closed-loop system with the Glucosym simulator and OpenAPS controller software keeps the BG in the target range for 93.49% of the time on average, demonstrating the advance of this PID-based control algorithm in diabetes treatment over the regular insulin therapy with Basal-Bolus control algorithm.

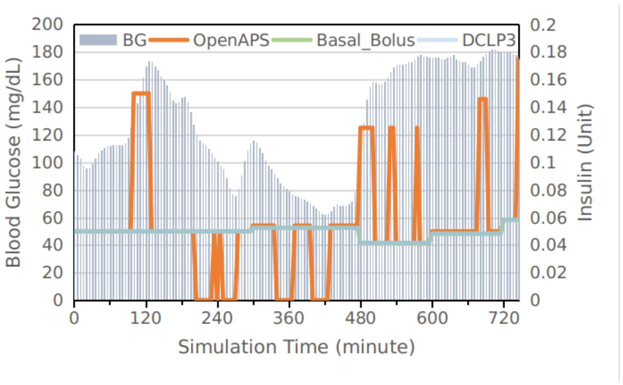

Figure 8: An Example of the Insulin Outputs among Each Controller Given the Same Glucose Readings.

The primary outcomes of each closed-loop APS and the actual control system in the clinical trial across six months are shown in Fig. 9. We see that the mean percentage of time with glucose values within the target range remained at a similar level during the six months in the clinical trial and both closed-loop simulations.

The closed-loop simulation offers a platform to evaluate or improve different pump algorithms with various patient profiles. For example, Fig. 10 shows the different decisions made by each control algorithm at each time step during 12.5 hours of treatment/simulation. We see that the actual pump from the clinical trial used a fixed basal rate and thus failed to keep the BG within the target range, resulting in adverse hyperglycemia. In comparison, the OpenAPS kept the BG value safe by increasing the insulin infusion when the BG is predicted to increase quickly and keeping a low insulin dose when the insulin on board is still at a high level after a large amount of previous insulin injection. Similarly, the Basal-Bolus controller with the UVA-Padova simulator issued a higher basal rate to avoid the BG value increasing too fast. Through such simulation and comparison, the proposed testbed can help to improve different control algorithms used in commercial insulin pumps and reduce patient harm or complaints.

4.4 Adverse Event Simulation

From Table 1 and Table 10, we see that adverse events naturally happen during clinical trials or home treatment of diabetic patients

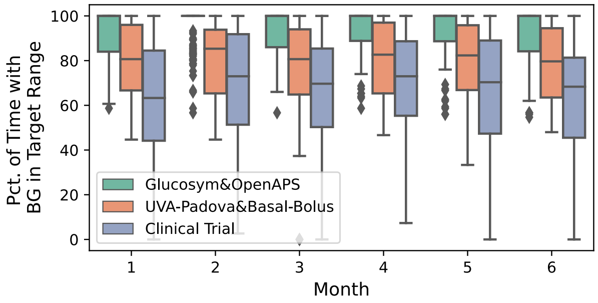

Figure 9: Percentage of Time with BG in the Target Range of 70 to 180 mg/dL for Clinical Train and Two Closed-loop APS.

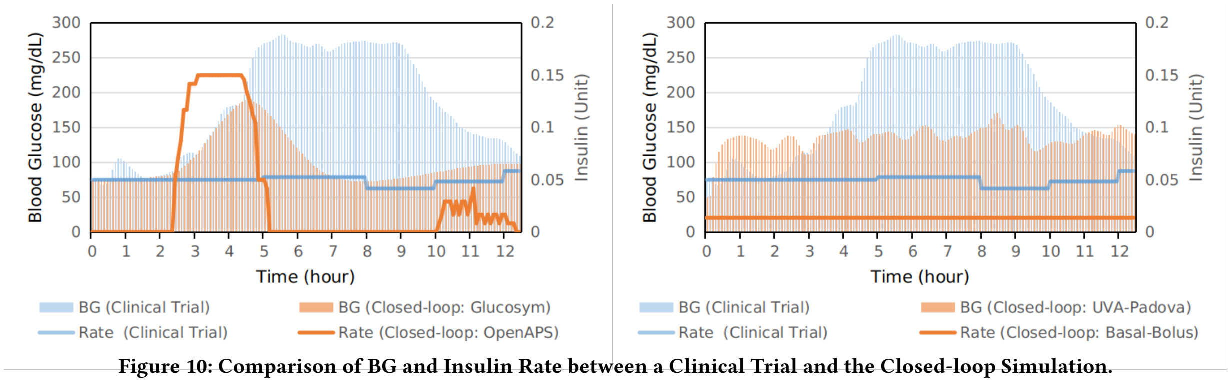

Figure 10: Comparison of BG and Insulin Rate between a Clinical Trial and the Closed-loop Simulation.

with APS due to device malfunctions or control software defects. However, it is expensive or risky for the manufacturers to test and improve the control algorithms through experiments on actual patients within realistic environments. The proposed closed-loop simulation offers an alternative way to evaluate the effectiveness of different control algorithms with actual patient profiles. However, the rate of adverse events in the closed-loop simulation is too low to evaluate the resilience of the target control algorithm comprehensively. Therefore, the adverse event simulator proposed in Section 3.4 was used to simulate the following scenarios: hyperglycemic adverse events (diabetic ketoacidosis), hypoglycemic events, malfunction of the device, and injury of patients.

We run 882 simulations (14 fault scenarios, as *Add/Sub* includes multiple sub-scenarios, times 9 random start times and durations times 7 initial BG values), and each simulation includes 12.5 hours of sensor data and insulin outputs after a meal with different carbohydrate inputs. An example of the BG trajectory with a fault injected starting at 400 minutes to simulate a CGM sensor reading malfunction is shown in Fig. 11. We see that the controller increased the insulin injection significantly based on the erroneous CGM readings while the BG value was not high, which further decreased the BG value under 50 mg/dL and resulted in a severe hypoglycemic event (marked by the red region in Fig. 11).

Our simulation generates two and a half years of data for 20 diabetes patients with different types of adverse events. The percentage of adverse events in the Glucosym simulator and UVA-Padova simulator are 33.9% and 39.3%, respectively (see Fig. 12). The generated synthetic dataset is available on Github [https://github.com/UVA-DSA/APS_TestBed].

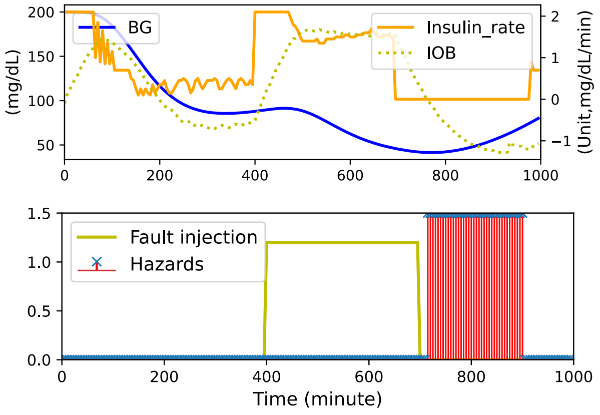

Figure 11: An Example of Simulated Hypoglycemic Adverse Event due to Fault Injection.

5 RELATED WORK

MCPS Testbeds: Testbeds are used in place of time- and resource-intensive clinical trials, so development has gravitated towards systems that affect the most critical organs. For instance, heart testbeds have been constructed for pacemaker validation [73] and cardiovascular interventions [71]. Robotic surgery testbeds have also been made for MRI-guided biopsy [65], endovascular surgery [29], and reconstructive surgeries in the hand [26]. In-silico trials of an insulin control algorithm are developed recently to facilitate research on APS [68, 68]. To the best of our knowledge, this paper is the first to develop an open-source closed-loop testbed for APS with real-world controllers, physical simulators of an extended set of patient profiles, and a realistic adverse event simulator.

Glucose Simulators: The use of simulators is vital in the development of APS. A glucose minimum model, a simple mathematical model for glucose levels, was first proposed in 1970 [36]. The UVA-Padova simulator [11] was the first APS simulator to be approved by the FDA in 2008 as a substitute for animal testing. A second simulation developed by a group at Cambridge University was released soon after in 2010, specifically targeted toward closed-loop APS simulation and virtual patient modeling [33]. The UVA-Padova was updated in 2014 [12] to improve the glucose kinetics model during hypoglycemia as well as incorporate glucagon kinetics and was accepted by the FDA as a substitute for certain preclinical trials. Glucosym, an open-source APS simulator, was released in 2017 to

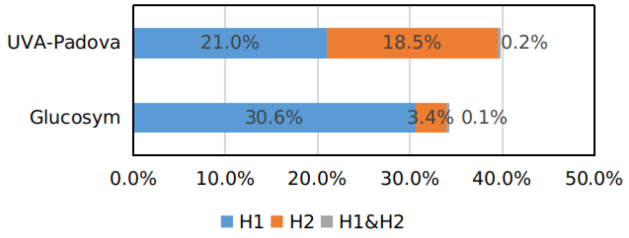

Figure 12: Success Rate of the Fault Injection Experiments in Simulating Low BG Hazard (H1) and High BG Hazard (H2), which may Result in Hypoglycemic or Hyperglycemic Adverse Event.

widen the availability of closed-loop APS simulation and testing [66]. In 2019, a group at the Oregon Health and Science University (OHSU) published an APS simulator based on a similar glucoregulatory model as the Cambridge simulator but with different insulin kinetics [35]. Our work differentiates from these previous works by integrating two advanced control software with the state-of-the-art simulators into a closed-loop APS testbed and proposing an optimization method to estimate real patient profiles for the closed-loop simulation.

Safety Evaluation of APS: Many previous works have also focused on evaluating the safety of APS control software, such as safety and effectiveness evaluation of insulin pump therapy in children and adolescents with Type 1 diabetes [2, 30], safety and efficacy review of commercial and emerging hybrid closed-loop systems [25, 58], generic safety requirements for developing safe insulin pump software [50] and insulin pump software certification [48], or safety evaluation of do-it-yourself APS [69]. However, most of these works relied on high-risk clinical tests or were not able to assess the resilience of tested insulin pumps against adverse events. In this paper, we integrate an adverse simulator into the closed-loop APS testbed, which could help with the evaluation of different APS control algorithms and safety mechanisms in preventing adverse events while avoiding actual harm to the patients.

6 CONCLUSION

Using two state-of-the-art glucose simulators, we develop a testbed for evaluating the performance of the control algorithms and safety features in APS. We assess the validity of the simulator by reverse-engineering the profiles of patients in a real clinical trial and demonstrating that the BG traces generated by each simulator are functionally the same as the BG traces from the trial. We also show the testbed's utility for closed-loop simulation by implementing two control algorithms to regulate the glucose levels of virtual patients. To push the APS to its limit, we embed a novel fault injection engine based on real FDA recalls into the testbed so performance can be evaluated in even the most hostile scenarios.

As research turns toward adopting more advanced data-driven methods like machine learning for the design of control algorithms, the proposed testbed and other *in silico* testing strategies will be essential for both final product evaluation and sourcing large quantities of high-quality data. This testbed can also be used to further develop personalized treatments by tailoring control algorithms to individual or similar patient profiles and to help diabetic patients understand their treatments by modeling their physiological dynamics. In future work, our testbed could be improved by developing more accurate estimation methods for patient profiles and incorporating meal and activity models and simulators.

ACKNOWLEDGMENT

This work was partially supported by the Commonwealth of Virginia under Grant CoVA CCI: C-Q122-WM-02 and by the National Science Foundation (NSF) under Grant No. 2146295.

REFERENCES

[1] Richard N. Bergman, Diane T. Finegood, Marilyn Ader. 1985. Assessment of Insulin Sensitivity in Vivo. *Endocrine Reviews* 6(1), 45–86.

[2] Ahlam Alotaibi, Reem Al Khalifah, and Karen McAssey. 2020. The efficacy and safety of insulin pump therapy with predictive low glucose suspend feature in decreasing hypoglycemia in children with type 1 diabetes mellitus: A systematic review and meta-analysis. *Pediatric Diabetes* 21, 7, 1256–1267.

[3] Mark A Atkinson, George S Eisenbarth, and Aaron W Michels. 2014. Type 1 diabetes. *The Lancet* 383, 9911, 69–82.

[4] Tandem Diabet Care. 2022. *Basal-IQ.* https://www.tandemdiabetes.com/providers/products/basal-iq

[5] OpenAPS community. 2017. *Understanding the determine-basal logic.* Retrieved June 3, 2021 from https://openaps.readthedocs.io/en/latest/

[6] Harry Emerson, Matt Guy, and Ryan McConville. 2022. *Offline Reinforcement Learning for Safer Blood Glucose Control in People with Type 1 Diabetes.* Technical Report. arXiv.

[7] Alvaro A. Cárdenas, Saurabh Amin, Lin Zong-Syun, et al. 2011. Attacks against process control systems: risk assessment, detection, and response. In *the 6th ACM Symposium on Information, Computer and Communications Security.* 355–366.

[8] Aaron Bray, Jeffrey B. Webb, Andinet Enquobahrie, et al. 2019. Pulse Physiology Engine: an Open-Source Software Platform for Computational Modeling of Human Medical Simulation. *SN Comprehensive Clinical Medicine* 1, 5, 362–377.

[9] Andrea Facchinetti, Simone Del Favero, Giovanni Sparacino, et al. 2014. Modeling the Glucose Sensor Error. *IEEE Transactions on Biomedical Engineering* 61, 620–629.

[10] Andreas Melmer, Thomas Züger, Dana M Lewis, et al. 2019. Glycaemic control in individuals with type 1 diabetes using an open source artificial pancreas system (OpenAPS). *Diabetes, Obesity and Metabolism* 21, 10, 2333–2337.

[11] Boris P. Kovatchev, Marc Breton, Chiara Dalla Man, et al. 2009. In Silico Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes. *Journal of Diabetes Science and Technology* 3, 1, 44–55.

[12] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, et al. 2014. The UVA/PADOVA Type 1 Diabetes Simulator. *Journal of Diabetes Science and Technology* 8, 1, 26–34.

[13] Charrise M Ramkissoon, Brian Aufderheide, B Wayne Bequette, et al. 2017. A review of safety and hazards associated with the artificial pancreas. *IEEE reviews in biomedical engineering* 10, 44–62.

[14] Divya Govindaraju, RR Shree, S Priyanka, et al. 2022. Event Detection and Control of Blood Glucose Levels Using Deep Neural Network. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems.* Springer, 235–245.

[15] David M. Maahs, Peter Calhoun, Bruce A. Buckingham, et al. 2014. A randomized trial of a home system to reduce nocturnal hypoglycemia in type 1 diabetes. *Diabetes Care* 37, 7, 1885–1891.

[16] Elena Chertok Shacham, Hila Kfir, Naama Schwartz, et al. 2018. Glycemic control with a basal-bolus insulin protocol in hospitalized diabetic patients treated with glucocorticoids: a retrospective cohort. *BMC Endocr Disord* 18(75), 1472–6823.

[17] Fraser Cameron, B Wayne Bequette, Darrell M Wilson, et al. 2011. A Closed-Loop Artificial Pancreas Based on Risk Management. *Journal of Diabetes Science and Technology* 5, 2, 368–379.

[18] Farooq Gessa, Philip Asare, Aaron Bray, et al. 2019. Towards a test and validation framework for closed-loop physiology management systems for critical and perioperative care. *ACM SIGBED Review* 16, 2, 31–40.

[19] Giuseppe Fico, Liss Hernández, Jorge Cancela, et al. 2017. Exploring the frequency domain of continuous glucose monitoring signals to improve characterization of glucose variability and of diabetic profiles. *Journal of diabetes science and technology* 11, 4, 773–779.

[20] Homa Alemzadeh, Catello Di Martino, Zhanpeng Jin, et al. 2012. Towards resiliency in embedded medical monitoring devices. In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops.* 1–6.

[21] Homa Alemzadeh, Daniel Chen, Xiao Li, et al. 2016. Targeted Attacks on Teleoperated Surgical Robots: Dynamic Model-Based Detection and Mitigation. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).* 395–406.

[22] Homa Alemzadeh, Daniel Chen, Zbigniew Kalbarczyk, et al. 2015. A software framework for simulation of safety hazards in robotic surgical systems. In *Special Issue on Medical Cyber Physical Systems Workshop*, Vol. 12. SIGBED Review.

[23] Homa Alemzadeh, Jaishankar Raman, Nancy Leveson, et al. 2016. Adverse events in robotic surgery: a retrospective study of 14 years of FDA data. *PloS one* 11, 4, e0151470.

[24] Homa Alemzadeh, Ravishankar K Iyer, Zbigniew Kalbarczyk, et al. 2013. Analysis of safety-critical computer failures in medical devices. *IEEE Security & Privacy* 11, 4, 14–26.

[25] Irl B Hirsch, Jill Abelseth, Bruce W Bode, et al. 2008. Sensor-augmented insulin pump therapy: results of the first randomized treat-to-target study. *Diabetes technology & therapeutics* 10, 5, 377–383.

[26] James A. Tigue, W. Bradford Rockwell, K. Bo Foreman, et al. 2020. Simulating Tendon Shortening During Flexor Tendon Repair Surgery Using A Biomechanical Model and Robotic Testbed. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob).* 193–198.

[27] John Walsh, Ruth Roberts, Timothy S. Bailey, et al. 2018. Bolus Advisors: Sources of Error, Targets for Improvement. *Journal of Diabetes Science and Technology* 12(1), 190–198.

[28] Kamuran Turksoy, Lauretta T. Quinn, Elizabeth Littlejohn, et al. 2014. An Integrated Multivariable Artificial Pancreas Control System. *Journal of Diabetes Science and Technology* 8, 3, 498–507.

[29] Lennart Karstensen, Torben Pätz, Franziska Mathis-Ullrich, et al. 2021. Facile Testbed for Robotic X-ray based Endovascular Interventions. *Current Directions in Biomedical Engineering* 7, 1, 121–125.

[30] Leslie P Plotnick, Loretta M Clark, Frederick L Brancati, et al. 2003. Safety and effectiveness of insulin pump therapy in children and adolescents with type 1 diabetes. *Clinical Diabetology* 4, 3, 245–252.

[31] Marzia Cescon, Daniel J DeSalvo, Trang T Ly, et al. 2016. Early detection of infusion set failure during insulin pump therapy in type 1 diabetes. *Journal of diabetes science and technology* 10, 6, 1268–1276.

[32] Marc D. Breton, Lauren G. Kanapka, Roy W. Beck, et al. 2020. A Randomized Trial of Closed-Loop Control in Children with Type 1 Diabetes. *The New England Journal of Medicine* 383, 9, 836–845.

[33] Malgorzata E. Wilinska, Ludovic J. Chassin, Carlo L. Acerini, et al. 2010. Simulation Environment to Evaluate Closed-Loop Insulin Delivery Systems in Type 1 Diabetes. *Journal of Diabetes Science and Technology* 4, 1, 132–144.

[34] Muhammad Taimoor Khan, Martin Pinzger, Dimitrios Serpanos, et al. 2020. Runtime Protection of Real-time Critical Control Applications against Known Threats. *IEEE Design & Test* 37, 6, 88–95.

[35] Navid Resalat, Joseph El Youssef, Nichole Tyler, et al. 2019. A statistical virtual patient population for the glucoregulatory system in type 1 diabetes with integrated exercise model. *PLOS ONE* 14, 7.

[36] PA Insel, KJ Kramer, RS Sherwin, et al. 1974. Modeling the insulin-glucose system in man. *Fed Proc* 33, 7, 1865–8.

[37] Po-Liang Wu, Dhashrath Raguraman, Lui Sha, et al. 2014. A Treatment Validation Protocol for Cyber-Physical-Human Medical Systems. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. 183–190.

[38] Qingnan Sun, Marko V Jankovic, Lia Bally, et al. 2018. Predicting blood glucose with an lstm and bi-lstm based deep neural network. In *2018 14th symposium on neural networks and applications (NEUREL)*. IEEE, 1–5.

[39] Rayhan A Lal, Laya Ekhlaspour, Korey Hood, et al. 2019. Realizing a Closed-Loop (Artificial Pancreas) System for the Treatment of Type 1 Diabetes. *Endocrine Reviews* 40, 6, 1521–1546.

[40] Revital Nimri, Natasa Bratina, Olga Kordonouri, et al. 2017. MD-Logic overnight type 1 diabetes control in home settings: A multicentre, multinational, single blind randomized trial. *Diabetes, Obesity and Metabolism* 19, 4, 553–561.

[41] R S Sherwin, K J Kramer, J D Tobin, et al. 1974. A Model of the Kinetics of Insulin in Man. *J Clin Invest* 53(5), 1481–1492.

[42] Roberto Visentin, Enrique Campos-Náñez, Michele Schiavon, et al. 2018. The UVA/Padova type 1 diabetes simulator goes from single meal to single day. *Journal of diabetes science and technology* 12, 2, 273–281.

[43] Sue A Brown, Boris P Kovatchev, Dan Raghinaru, et al. 2019. Six-month randomized, multicenter trial of closed-loop control in type 1 diabetes. *New England Journal of Medicine* 381, 18, 1707–1717.

[44] Sunil Deshpande, Jordan E. Pinsker, Stamatina Zavitsanou, et al. 2019. Design and Clinical Evaluation of the Interoperable Artificial Pancreas System (iAPS) Smartphone App: Interoperable Components with Modular Design for Progressive Artificial Pancreas Research and Development. *Diabetes Technology & Therapeutics* 21, 1, 35–43.

[45] SS Kanderian, S Weinzimer, G Voskanyan, et al. 2009. Identification of Intraday Metabolic Profiles during Closed-Loop Glucose Control in Individuals with Type 1 Diabetes. *Journal of diabetes science and technology* 3, 5, 1047–1057.

[46] Sediqeh Samadi, Mudassir Rashid, Kamuran Turksoy, et al. 2018. Automatic Detection and Estimation of Unannounced Meals for Multivariable Artificial Pancreas System. *Diabetes Technology & Therapeutics* 20, 3, 235–246.

[47] Xugui Zhou, Anna Schmedding, Haotian Ren, et al. 2022. Strategic Safety-Critical Attacks Against an Advanced Driver Assistance System. In *2022 52nt Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*.

[48] Yihai Chen, Mark Lawford, Hao Wang, et al. 2013. Insulin pump software certification. In *International Symposium on Foundations of Health Informatics Engineering and Systems*. Springer, 87–106.

[49] Yanchen Xu, Daniel Tran, Yuan Tian, et al. 2019. Analysis of cyber-security vulnerabilities of interconnected medical devices. In *2019 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*.

[50] Yi Zhang, Raoul Jetley, Paul L Jones, et al. 2011. Generic safety requirements for developing safe insulin pump software. *Journal of diabetes science and technology* 5, 6, 1403–1419.

[51] U.S. Food and Drug Administration. 2020. *Recalls, Corrections and Removals (Devices)*. Retrieved September 29, 2020 from https://www.fda.gov/medical-devices/postmarket-requirements-devices/recalls-corrections-and-removals-devices

[52] US Food and Drug Administration. 2022. *CFR - Code of Federal Regulations Title 21*. Retrieved Mar 29, 2022 from https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/cfrsearch.cfm?fr=312.32

[53] U.S. Food and Drug Administration. 2022. *MAUDE - Manufacturer and User Facility Device Experience*. Retrieved June 3, 2022 from https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfMAUDE/search.cfm

[54] U.S. Food and Drug Administration. 2022. *MAUDE Adverse Event Report*. Retrieved June 3, 2022 from https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfMAUDE/Detail.cfm?MDRFOI__ID=2417117

[55] U.S. Food and Drug Administration. 2022. *MAUDE Adverse Event Report*. Retrieved June 3, 2022 from https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfMAUDE/Detail.cfm?MDRFOI__ID=2430675

[56] U.S. Food and Drug Administration. 2022. *MAUDE Adverse Event Report*. Retrieved June 3, 2022 from https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfMAUDE/detail.cfm?mdrfoi__id=6113784&pc=LZG

[57] U.S. Food and Drug Administration. 2022. *Medical Device Recalls*. Retrieved June 3, 2022 from https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfres/res.cfm

[58] Julia Fuchs and Roman Hovorka. 2020. Closed-loop control in insulin pumps for type-1 diabetes mellitus: safety and efficacy. *Expert review of medical devices* 17, 7, 707–720.

[59] Richard Hawkins, Colin Paterson, and Chiara Picardi et al. 2021. Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS).

[60] Ruiqiang Hu and Chengwei Li. 2015. An Improved PID Algorithm Based on Insulin-on-Board Estimate for Blood Glucose Control with Type 1 Diabetes. *Computational and Mathematical Methods in Medicine*.

[61] Allen B. King and Dana U. Armstrong. 2007. A Prospective Evaluation of Insulin Dosing Recommendations in Patients with Type 1 Diabetes at Near Normal Glucose Control: Bolus Dosing. *Journal of diabetes science and technology (Online)* 1, 1, 42–46.

[62] Dana Lewis, Scott Leibrand, and # OpenAPS Community. 2016. Real-world use of open source artificial pancreas systems. *Journal of diabetes science and technology* 10, 6, 1411–1411.

[63] Dana M Lewis. 2020. Do-it-yourself artificial pancreas system and the OpenAPS movement. *Endocrinology and Metabolism Clinics* 49, 1, 203–213.

[64] Chunxiao Li, Anand Raghunathan, and Niraj K Jha. 2011. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*. IEEE, 150–156.

[65] Evelyn Mendoza and John Peter Whitney. 2019. A Testbed for Haptic and Magnetic Resonance Imaging-Guided Percutaneous Needle Biopsy. *IEEE Robotics and Automation Letters* 4, 4, 3177–3183.

[66] Perceptus. 2017. *GlucoSym*. Retrieved May 3, 2021 from https://github.com/Perceptus/GlucoSym

[67] Aria Saunders, Laurel H. Messer, and Gregory P. Forlenza. 2019. MiniMed 670G Hybrid Closed Loop Artificial Pancreas System for the treatment of type 1 diabetes mellitus: overview of its safety and efficacy. *Expert review of medical devices* 16, 10, 845–853.

[68] Jana Schmitzer, Carolin Strobel, Ronald Blechschmidt, Adrian Tappe, and Heiko Peuscher. 2022. Efficient closed loop simulation of do-it-yourself artificial pancreas systems. *Journal of Diabetes Science and Technology* 16, 1 (2022), 61–69.

[69] Chiara Toffanin, Milos Kozak, Zdenek Sumnik, Claudio Cobelli, and Lenka Petruzelkova. 2020. In silico trials of an open-source android-based artificial pancreas: a new paradigm to test safety and efficacy of do-it-yourself systems. *Diabetes technology & therapeutics* 22, 2 (2020), 112–120.

[70] Phuwadol Viroonluecha, Esteban Egea-Lopez, and Jose Santa. 2021. Evaluation of Blood Glucose Level Control in Type 1 Diabetic Patients Using Deep Reinforcement Learning.

[71] Gustaaf J. Vrooijink, Hassna Irzan, and Sarthak Misra. 2018. A Beating Heart Testbed for the Evaluation of Robotic Cardiovascular Interventions. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. 1076–1082.

[72] Dyanne P Westerberg. 2013. Diabetic ketoacidosis: evaluation and treatment. *American family physician* 87, 5, 337–346.

[73] Zhihao Jiang, A Connolly, and R Mangharam. 2010. Using the Virtual Heart Model to validate the mode-switch pacemaker operation. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. 6690–6693.

[74] Xugui Zhou, Bulbul Ahmed, and James H Aylor et al. 2021. Data-driven design of context-aware monitors for hazard prediction in artificial pancreas systems. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 484–496.

[75] Xugui Zhou, Maxfield Kouzel, and Homa Alemzadeh. 2022. Robustness Testing of Data and Knowledge Driven Anomaly Detection in Cyber-Physical Systems. *5th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop on Dependable and Secure Machine Learning (DSN-DSML)* (2022).

[76] Taiyu Zhu, Kezhi Li, and Pantelis Georgiou. 2021. Personalized Dual-Hormone Control for Type 1 Diabetes Using Deep Reinforcement Learning. In *Explainable AI in Healthcare and Medicine*. Vol. 914. Springer International Publishing, Cham, 45–53.