НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
(повна назва інституту/факультету)
кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ
(повна назва кафедри)

«До захисту допущено»
Завідувач кафедри БМК

_____ Світлана АЛХІМОВА
(підпис) (ініціали, ПРІЗВИЩЕ

"___" __*червня*__ 2025р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп'ютерні технології в біології та медицині»
зі спеціальності *122 «Комп'ютерні науки»*

на тему: Програмний застосунок для симуляції регуляції глюкози у пацієнтів із діабетом першого типу

Виконав: студент **IV** курсу, групи _БС-15__

ГЕРЦЕНБЕРГ ВЛАДИСЛАВ МАКСИМОВИЧ _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник: *професор каф. біомедичної кібернетики доц., д.т.н., Зеленський Кирило Харитонович* _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по ініціали) (підпис)

Рецензент: *доцент каф. біомедичної інженерії, доц., к.т.н. Богомолов Микола Федорович* _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут (факультет) БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
(повна назва)

Кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ
(повна назва)

Рівень вищої освіти – перший (бакалаврський)
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри БМК
_____ Світлана АЛХІМОВА
«___»__*квітень*___2025 р.

ЗАВДАННЯ
на дипломну роботу студенту

ГЕРЦЕНБЕРГУ ВЛАДИСЛАВУ МАКСИМОВИЧУ
(прізвище, ім'я, по батькові)

1. Тема роботи **Програмний застосунок для симуляції регуляції глюкози у пацієнтів із діабетом першого типу**

керівник роботи
Зеленський Кирило Харитонович, д.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «26» *травня* 2025 р. ***№1758-с***

2. Термін подання студентом роботи ***22-23 травня 2025 року***
3. Вихідні дані: Дані з наукових публікацій щодо рівня глюкози у пацієнтів із діабетом 1 типу. Для моделювання використовуються диференціальні рівняння.
4. Зміст роботи: Проаналізувати літературу та існуючі рішення з моделювання регуляції глюкози у пацієнтів із діабетом 1 типу. Провести обґрунтування та вибір методів математичного моделювання, зокрема диференціальних рівнянь. Розробити та протестувати програмний продукт для симуляції змін рівня глюкози.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): 15 слайдів *(презентація з захисту ДР)*.
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Дипломної роботи			

7. Дата видачі завдання ***12 травня 2025р.***

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 24.12.2024р.	***виконано***
2	Переддипломна практика. Виконання практичної частини ДР та додаткових розділів	1-4 тиждень за графіком практики	***виконано***
3	Виконання основних розділів ДР (Вступ, програмна реалізація та методика роботи програми для симуляції регуляції глюкози)	Протягом усієї практики	***виконано***
4	Перевірка ДР науковим керівником	до 20.05.2025	***виконано***
5	Подання в електронному вигляді ДР на перевірку нормоконтролера та плагіат (StrikePlagiarism.com). Доопрацювання ДР	Не пізніше 23.05.2025	***виконано***
6	Отримати відгук від керівника ДР	Не пізніше 05.06.2025	***виконано***
7	Надати пакету документів по ДР та супровідних до неї документів на засідання кафедри		***виконано***
8	Подання ДР рецензенту. Отримання рецензії.	09.06 – 13.06.2025	***виконано***
9	Захист ДР в ЕК	*16.06 - -21.06.2025*	

Студент _____ **Владислав ГЕРЦЕНБЕРГ**
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник ДР _____ **Кирило ЗЕЛЕНСЬКИЙ**
(підпис) (ім'я, ПРІЗВИЩЕ)

Нормоконтролер _____ **Галина КОРНІЄНКО**
(підпис) (ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Дипломна робота за темою «Програмний застосунок для симуляції регуляції глюкози у пацієнтів із діабетом першого типу» виконана студентом кафедри біомедичної кібернетики ФБМІ *Герценбергом Владиславом Максимовичем зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та* складається зі: вступу; 5 розділів, висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 40 джерел, та додатків. Загальний обсяг роботи 63 сторінок.

Актуальність теми. 80% хворих на ЦД-1 регулярно стикаються з небезпечними гіпо-/гіперглікеміями, що підвищує ризик ускладнень. Системи «штучна підшлункова» знижують варіабельність глюкози на 20-30 %, однак потребують точних симуляторів для налаштування алгоритмів. Розробка Python-платформи для моделювання та розрахунку оптимальних доз інсуліну сприятиме удосконаленню клінічних протоколів і зменшенню вартості випробувань.

Мета і завдання роботи.

Мета роботи полягає у поглибленні розуміння процесів регуляції глюкози в крові при цукровому діабеті 1 типу через розробку програмного інструменту, який моделює динаміку «глюкоза – інсулін» з урахуванням прийомів їжі та демонструє можливості зовнішнього PD-контролера для підтримання глікемії в цільових межах.

Її досягнення передбачає вирішення таких завдань:

- аналітичний огляд вітчизняних і зарубіжних джерел щодо математичних моделей глікемії та автоматизованих систем інсулінотерапії;
- вибір і адаптацію моделі Бергмана для багатопацієнтного використання;

- розробку PD-алгоритму дозування інсуліну з урахуванням порогів безпеки;
- створення крос-платформеного Python-застосунку з графічним інтерфейсом Tkinter, візуалізацією результатів і модулем логування;
- проведення серії симуляцій для «здорових» і діабетичних профілів (із контролером / без контролера) та порівняння основних глікемічних метрик;
- техніко-економічне обґрунтування вибору архітектури та оцінку собівартості розробки.)

Використані методи. Функціонально-вартісний аналіз; математичне моделювання системою ЗДР; мова програмування Python; бібліотеки SciPy, Pandas, Matplotlib, NumPy, Tkinter.

Отримані результати. Створено та протестовано працюючий прототип «штучної підшлункової залози», який моделює 24-годинний профіль глюкози менш ніж за 30 с, дозволяє обирати один із 4 пацієнтських пресетів, автоматично зберігає графіки та метрики, а також демонструє зниження пікової гіперглікемії діабетичного пацієнта з ≈ 15 до 8 ммоль/л при ввімкненні PD-контролера. Собівартість розробки за оптимальним варіантом становить ≈ 400 тис. грн, інтегральний показник техніко-економічного рівня $К_{ТЕР} = 1{,}26 \cdot 10^{-4}$ підтверджує доцільність обраної конфігурації. Розробка придатна для навчального процесу й подальшої інтеграції з реальними CGM-даними.

Публікації. Не заплановано.

Ключові слова. цукровий діабет 1 типу, симуляція глікемії, математична модель Бергмана, PD-контролер інсуліну, штучна підшлункова залоза, Python-застосунок, Tkinter-GUI, автоматизована інсулінотерапія.

Бібліографічний опис ДР

Герценберг В. М. Програмний застосунок для симуляції регуляції глюкози у пацієнтів із діабетом першого типу: дипломна роб. бакалавра: 122 Комп’ютері науки / Герценберг Владислав Максимович. – Київ, 2025. – 80 с.

ANNOTATION

Bachelor's thesis on "Software application for simulating glucose regulation in patients with type 1 diabetes" has been prepared by Hertsenberh Vladyslav, student of the Department of Biomedical Cybernetics, Faculty of Biomedical Engineering, speciality 122 "Computer Science", educational-professional programme "Computer Technologies in Biology and Medicine". The thesis comprises: an introduction; five chapters with individual conclusions; overall conclusions; a reference list of 40 sources; and appendices. Total length – 63 pages.

Relevance of the topic. About 80 % of people with T1D regularly experience hazardous episodes of hypo- and hyperglycaemia, markedly increasing the risk of acute and chronic complications. Closed-loop "artificial pancreas" systems can reduce glucose variability by 20–30 %, but their control algorithms must first be tuned and verified in high-fidelity simulators. Developing a Python-based platform for modelling glucose–insulin dynamics and calculating optimal insulin doses therefore facilitates refinement of clinical protocols and lowers the cost of pre-clinical trials.

Purpose and objectives.

The purpose of this work is to deepen the understanding of glucose-insulin regulation in type 1 diabetes by developing a software tool that models glucose dynamics with meal disturbances and demonstrates how an external PD controller can maintain blood glucose within target ranges.

To achieve this goal, the following tasks were solved:

- Analytical survey of Ukrainian and international sources on mathematical glucose models and automated insulin-delivery systems;
- Selection and adaptation of the Bergman minimal model for multi-patient use;
- Development of a safety-aware PD dosing algorithm;
- Implementation of a cross-platform Python application with a Tkinter

GUI, result visualisation and logging;

- Execution of simulation series for “healthy” and diabetic profiles (controller on/off) and comparison of key glycaemic metrics;
- Techno-economic substantiation of the chosen architecture and cost estimation of the software product.

Methods. Functional-cost analysis; mathematical modelling with ordinary differential equations; Python programming language; SciPy, Pandas, Matplotlib, NumPy and Tkinter libraries.

Results obtained. A working prototype of an “artificial pancreas” was created and tested. It simulates a 24-hour glucose profile in under 30 s, allows selection of four patient presets, automatically saves plots and metrics, and shows that switching on the PD controller lowers the diabetic peak glucose from ≈ 15 to 8 mmol L^{-1}. The optimal development scenario costs about 400 k UAH; the integrated techno-economic index $К_{ТЕР} = 1{,}26 \cdot 10^{-4}$ confirms the feasibility of the chosen configuration. The software is suitable for educational use and can be further linked to real CGM data.

Publications. Not planned.

Key words. type 1 diabetes, glycaemia simulation, Bergman mathematical model, PD insulin controller, artificial pancreas, Python application, Tkinter GUI, automated insulin delivery.

Bibliographic reference

Hertsenberh V. M. Software Application for Simulation of Glucose Regulation in Patients with Type 1 Diabetes: Bachelor’s thesis / Hertsenberh Vladyslav Maksymovych. – Kyiv, 2025. – 80 p.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 10

ВСТУП 12

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РЕГУЛЯЦІЇ ГЛЮКОЗИ ТА АВТОМАТИЗОВАНОГО ВВЕДЕННЯ ІНСУЛІНУ ПРИ ЦД 1 ТИПУ 14

1.1 Актуальність проблеми контролю глікемії при ЦД 1 типу ... 14

1.2 Фізіологічні основи гомеостазу глюкози 17

1.3 Клінічні особливості та наслідки цукрового діабету 1 типу ... 19

1.4 Сучасні підходи до моделювання системи «глюкоза – інсулін» 22

1.5 Автоматизовані системи інсулінотерапії (AID-системи) 25

Висновок з розділу 1 29

РОЗДІЛ 2 ОГЛЯД СУЧАСНИХ МОДЕЛЕЙ ТА ПРОГРАМНИХ ЗАСОБІВ МОДЕЛЮВАННЯ РЕГУЛЯЦІЇ ГЛЮКОЗИ ПРИ ЦД 1 ТИПУ 31

2.1 Огляд існуючих симуляторів та ПЗ для моделювання діабету 1 типу 31

2.2 Порівняльна характеристика функціоналу 35

2.3 Обґрунтування вибору власного підходу 38

Висновок з розділу 2 41

РОЗДІЛ 3 ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИМУЛЯЦІЇ РЕГУЛЯЦІЇ ГЛЮКОЗИ ПРИ ЦД 1 ТИПУ 42

3.1 Мова програмування та середовище розробки 43

3.2 Бібліотеки та інструменти для математичного моделювання 44

3.3 Засоби реалізації інтерфейсу та візуалізації 50

3.4 Вимоги до апаратного та програмного забезпечення 51
Висновок до розділу 3 53
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА РОБОТИ ПРОГРАМИ ДЛЯ СИМУЛЯЦІЇ РЕГУЛЯЦІЇ ГЛЮКОЗИ У ПАЦІЄНТІВ ІЗ ДІАБЕТОМ ПЕРШОГО ТИПУ 55
4.1 Архітектура Python-додатку для моделювання глікемії 55
4.2 Реалізація математичної моделі 57
4.3 Реалізація PD-контролера та логіка Safety 59
4.4 Компоненти графічного інтерфейсу 62
4.5 Моделювання сценаріїв та результати експериментів 64
4.6 Розрахунок економічного ефекту за темою дипломною роботи 66
Висновок до розділу 4 68
РОЗДІЛ 5 ЗАХИСТ ІНФОРМАЦІЇ 70
5.1 Конфіденційність та цілісність даних 70
5.2 Захист від несанкціонованого доступу та зловмисних змін 71
5.3 Забезпечення надійності та відмовостійкості 72
5.4 Криптографічний захист 73
Висновок до розділу 5 73
ЗАГАЛЬНІ ВИСНОВКИ 74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД - база даних, структурована колекція даних, що зберігається та управляється комп'ютерною системою.

CSV - Comma-Separated Values, текстовий формат таблиць, у якому поля розділяються комами.

CISA - Cybersecurity and Infrastructure Security Agency (Агентство з кібербезпеки та інфраструктурної безпеки США).

AID - Automated Insulin Delivery, автоматизована система подачі інсуліну (штучна підшлункова)

D(t) - швидкість надходження глюкози в кров із їжі в момент часу t, mg/dL/min.

Gb - базальний (нормальний) рівень глюкози в плазмі, mg/dL.

GUI - Graphical User Interface, графічний інтерфейс користувача.

Ib - базальний рівень інсуліну в плазмі, μU/mL.

mg/dL - міліграми на децилітр (одиниця вимірювання концентрації глюкози).

min, хв - хвилина (одиниця часу).

MPC - Model Predictive Control, метод прогнозно-упереджувального керування.

ODU (укр. ОДР) - звичайні диференціальні рівняння (ordinary differential equations).

PD-контролер - пропорційно-диференційний регулятор (proportional-derivative controller).

PID-контролер - пропорційно-інтегрально-диференційний регулятор.

SSL/TLS - Secure Sockets Layer / Transport Layer Security, криптографічні протоколи захищеного передавання даних.

ЦД1 - Цукровий діабет 1 типу — автоімунне захворювання з

абсолютним дефіцитом інсуліну

HbA1c - глікований гемоглобін, інтегральний показник середньої глікемії за ≈ 90 днів

YAML - YAML Ain't Markup Language, формат ієрархічних текстових конфігураційних файлів.

ВСТУП

Актуальність теми даної роботи полягає у вирішенні проблеми регуляції рівня глюкози в крові у пацієнтів з цукровим діабетом 1 типу, стан яких характеризується абсолютною недостатністю інсуліну внаслідок аутоімунного руйнування бета-клітин підшлункової залози. В умовах відсутності природного вироблення інсуліну, підтримка рівня глюкози в крові в межах нормоглікемії (~5 ммоль/л) є критично важливим для запобігання гострим та хронічним ускладненням діабету

Створення програмного застосунку для моделювання динаміки глюкози та інсуліну в організмі пацієнтів з діабетом 1 типу може допомогти у розробці та оптимізації стратегій глікемічного контролю, зокрема із застосуванням зовнішніх регуляторів.

Мета і завдання роботи

Мета роботи полягає у покращенні розуміння процесів регуляції глюкози в крові при діабеті 1 типу шляхом розробки програмного інструменту для моделювання динаміки глюкози та інсуліну з урахуванням прийомів їжі та застосування зовнішнього PD-контролера для підтримки рівня глюкози в цільових межах.

Сформульовано наступні задачі:

1. Розробити математичну модель динаміки глюкози та інсуліну в організмі пацієнта з діабетом 1 типу, що враховує прийоми їжі.
2. Реалізувати зовнішній пропорційно-диференційний (PD) контролер для автоматичного регулювання введення інсуліну з метою підтримки рівня глюкози в заданому діапазоні.
3. Створити зручний графічний інтерфейс користувача (GUI) для проведення симуляцій та експериментів з різними параметрами моделі та контролера.
4. Провести серію симуляцій для демонстрації відмінностей між

глікемічними профілями здорової людини та пацієнта з діабетом 1 типу.

5. Оцінити ефективність розробленого PD-контролера у зниженні пікових значень глюкози після прийомів їжі за допомогою графічного відображення результатів симуляцій.

Використані методи. Функціонально-вартісний аналіз; математичне моделювання системою звичайних диференційних рівнянь; мова програмування Python; бібліотеки SciPy, NumPy, Pandas, Matplotlib та Tkinter; методи графічної візуалізації даних.

Отримані результати. Створено працездатний прототип «штучної підшлункової залози», що моделює 24-годинний профіль глюкози менш ніж за 30 с, дає змогу обирати один із 4 пацієнтських пресетів, автоматично зберігає графіки та метрики й демонструє зниження пікової гіперглікемії діабетичного пацієнта з ≈ 15 до 8 ммоль/л при ввімкненні PD-контролера. Собівартість розробки за оптимальним варіантом становить ≈ 400 тис. грн; інтегральний показник техніко-економічного рівня $К_{ТЕР} = 1{,}26 \cdot 10^{-4}$ підтверджує доцільність обраної конфігурації. Розробка придатна для навчального процесу та подальшої інтеграції з реальними CGM-даними.

Публікації. Не заплановано.

Структура роботи

Дипломна робота за темою «Програмний застосунок для симуляції регуляції глюкози у пацієнтів із діабетом першого типу» виконана студентом *Герценбергом Владиславом Максимовичем зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині»*, побудована за класичним типом та викладена на 78 сторінках машинописного тексту. Вона складається з: вступу; 5 розділів, висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 40 джерел та додатків (0 – на кирилиці, 40 – на латиниці). В роботі представлено 5 рисунків і 2 таблиці.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ РЕГУЛЯЦІЇ ГЛЮКОЗИ ТА АВТОМАТИЗОВАНОГО ВВЕДЕННЯ ІНСУЛІНУ ПРИ ЦД 1 ТИПУ

У цьому розділі подано теоретичний фундамент, що лежить в основі подальшої розробки симуляційного застосунку. Спочатку окреслено глобальну епідеміологію та медико-соціальні ризики цукрового діабету 1 типу, підкреслюючи критичну потребу в строгому контролі глікемії. Далі стисло описано фізіологічні механізми регуляції глюкози у здорової людини — збалансовану дію інсуліну та глюкагону, роль підшлункової залози й тканин-мішеней. Наступний підрозділ присвячено патофізіології ЦД1, типових коливань глюкози та їх гострих і хронічних ускладнень, що визначають клінічну потребу в інсулінотерапії. Окрему увагу приділено огляду математичних моделей системи «глюкоза – інсулін» — від мінімальної моделі Бергмана до UVa/Padova — із критичною оцінкою їх застосовності, спрощень та обмежень. Завершує розділ аналітичний огляд сучасних автоматизованих систем інсулінотерапії (AID): безперервний моніторинг глюкози, інсулінові помпи та алгоритми PID/MPC-керування, які формують практичний контекст для розробки власного PD-контролера.

1.1 Актуальність проблеми контролю глікемії при ЦД 1 типу

Цукровий діабет 1 типу (ЦД1) – це хронічне автоімунне захворювання, при якому відбувається руйнування β-клітин підшлункової залози, що продукують інсулін. Унаслідок цього виникає абсолютний дефіцит інсуліну, життєво необхідного гормону, який забезпечує надходження глюкози з крові в клітини та підтримує нормальний енергетичний обмін. ЦД1 становить приблизно 5–10% всіх випадків діабету (переважно інша частка припадає на

діабет 2 типу) [21]. Попри відносно меншу поширеність порівняно з діабетом 2 типу, ЦД1 здебільшого вражає дітей та молодь і потребує довічної інсулінотерапії, що робить проблему контролю глікемії надзвичайно актуальною [1].

За даними Міжнародної діабетичної федерації (IDF), у світі налічується понад 1,2 мільйона дітей та підлітків (до 20 років) з ЦД1, і цей показник щорічно зростає приблизно на 3%. Зокрема, у країнах Європи та Північної Америки спостерігається неухильне підвищення захворюваності на ЦД1 серед дітей у другій половині XX – на початку XXI століття, що підтверджує глобальну тенденцію до зростання випадків ЦД1 [12]. ЦД1 має тяжкі короткострокові та довгострокові наслідки для здоров'я, якщо глікемічний контроль незадовільний. Гострою небезпекою є розвиток діабетичного кетоацидозу (ДКА) при відсутності інсуліну – стану, який може швидко призвести до коми та смерті. Також гіпоглікемія (надмірне зниження рівня глюкози крові) є гострим ускладненням інсулінотерапії, що становить ризик для життя і обмежує агресивність лікування [1].

Хронічна гіперглікемія при недостатньому контролі ЦД1 спричиняє розвиток мікросудинних і макросудинних ускладнень. До мікросудинних ускладнень належать діабетична ретинопатія, нефропатія та нейропатія, що можуть призводити відповідно до сліпоти, ниркової недостатності та ампутацій кінцівок. Діабетична ретинопатія є найчастішим довгостроковим ускладненням: за оцінками, протягом життя ознаки ретинопатії розвинуться у 95–97% пацієнтів з ЦД1. Близько 3–8% хворих зрештою втрачають зір на одне чи обидва ока внаслідок діабетичних уражень сітківки. Ці факти роблять ЦД1 провідною причиною нових випадків сліпоти у працездатному віці в розвинених країнах [14]. Макросудинні ускладнення (атеросклероз судин серця, мозку, периферичних артерій) також часто розвиваються при тривалому перебігу ЦД1, підвищуючи ризик інфарктів та інсультів.

Результати багаторічних досліджень переконливо довели, що оптимальний контроль глікемії суттєво знижує ризик розвитку ускладнень ЦД1. Знакове дослідження DCCT (Diabetes Control and Complications Trial) показало, що інтенсивна інсулінотерапія, спрямована на підтримання рівня глюкози максимально близько до норми, зменшує частоту мікросудинних ускладнень (ретинопатії, нефропатії, нейропатії) до 50–76% порівняно зі стандартним лікуванням. Через 6,5 років після початку DCCT різниця в середньому рівні HbA1c між групами інтенсивного та звичайного лікування становила близько 2%, що й обумовило значне розходження в частоті ускладнень. Ба більше, подальше спостереження за учасниками (дослідження EDIC) показало, що переваги раннього інтенсивного контролю глікемії зберігались протягом десятиліття, явище так званої «метаболічної пам'яті». Також було продемонстровано тенденцію до зменшення макросудинних подій (інфарктів, інсультів) у групі з кращим контролем, особливо при довготривалому спостереженні [21]. Отже, підтримання цільових рівнів глюкози крові у хворих на ЦД1 прямо пов'язане з профілактикою інвалідизуючих ускладнень і покращенням прогнозу.

Водночас досягнення належного контролю глікемії в реальній клінічній практиці залишається складним завданням. Рекомендованою міжнародними настановами метою лікування є рівень глікованого гемоглобіну HbA1c < 7% у більшості дорослих пацієнтів з ЦД1 (та <7,5% у дітей) для мінімізації ризику ускладнень. Однак, великі реєстрові дослідження свідчать, що лише приблизно 20–30% пацієнтів з ЦД1 досягають цього цільового рівня HbA1c, тоді як у решти фіксуються хронічно підвищені показники глікемії. Зокрема, за даними американського реєстру T1D Exchange, близько 79% дорослих з ЦД1 мали HbA1c вище 7% [32]. Основними причинами труднощів у контролі є варіабельність потреби в інсуліні, ризик гіпоглікемій при інтенсифікації терапії, вплив поведінкових і психосоціальних чинників, особливо у молоді [10]. Значну роль відіграє також людський фактор – дотримання пацієнтами режиму інсулінотерапії, самоконтролю, дієти. Постійний самоконтроль є

великим тягарем для хворих: традиційна терапія вимагає частих вимірювань глюкози (4–8 разів на добу глюкометром або постійний моніторинг), багатьох ін'єкцій інсуліну чи носіння інсулінової помпи, розрахунку доз на вуглеводи тощо. Все це суттєво знижує якість життя пацієнтів і підлітків, може призводити до «вигорання» від діабету та помилок у терапії [1].

Таким чином, проблема досягнення безпечного та ефективного контролю глікемії при ЦД1 залишається актуальною як з клінічної, так і з соціально-економічної точки зору. Необхідні новітні підходи, що полегшать ведення хвороби та мінімізують ризики ускладнень. За останні десятиліття відбувся значний прогрес у технологіях лікування ЦД1 – зокрема, розробка систем безперервного моніторингу глюкози, інсулінових помп та алгоритмів автоматичного введення інсуліну [17]. У наступних підрозділах розглянуто фізіологічні основи регуляції глюкози в організмі, патофізіологію та наслідки ЦД1, сучасні підходи до моделювання системи «глюкоза – інсулін», а також принципи побудови автоматизованих інсулінових систем (AID-систем), що покликані вирішити проблему оптимального контролю глікемії.

1.2 Фізіологічні основи гомеостазу глюкози

Гомеостаз глюкози – це підтримання рівня глюкози крові в межах вузького фізіологічного діапазону, незважаючи на коливання вживання їжі чи фізичного навантаження. У здорових людей натще глікемія становить в середньому 4,4–5,0 ммоль/л (80–90 мг/дл), після прийому їжі підвищується до ~7,0–7,8 ммоль/л (120–140 мг/дл) і повертається до норми впродовж приблизно 2 годин завдяки швидкому фізіологічному регулюванню [1]. Ключову роль у цьому регулюванні відіграють гормони підшлункової залози – інсулін та глюкагон, які мають протилежну дію [25].

У підшлунковій залозі є острівці Лангерганса, β-клітини яких секретують інсулін, а α-клітини – глюкагон. Інсулін виділяється у відповідь на підвищення рівня глюкози крові (наприклад, після вуглеводного

навантаження). Він діє як анаболічний гормон: сприяє поглинанню глюкози клітинами м’язів і жирової тканини (активує транспортери GLUT4 у клітинній мембрані), стимулює печінку до синтезу глікогену (полімеризації надлишку глюкози в запасний вуглевод), пригнічує вихід глюкози з печінки, а також стимулює синтез білків і жирів. Глюкагон, навпаки, секретується при зниженні рівня глюкози (у періоди голодування чи між прийомами їжі) і реалізує катаболічну дію: стимулює розщеплення глікогену в печінці (глікогеноліз) і синтез глюкози з не-вуглеводних субстратів (глюконеогенез), що приводить до підвищення рівня глюкози в крові [25]. Інсулін і глюкагон взаємодіють за принципом негативного зворотного зв’язку, забезпечуючи відносну стабільність глікемії в межах ~3,5–8 ммоль/л у різних умовах [1]. Наприклад, після їди інсулін секреція зростає, що знижує глюкозу до норми, а при тривалому голодуванні посилюється секреція глюкагону, і печінка викидає глюкозу в кров для запобігання гіпоглікемії.

Окрім інсуліну та глюкагону, в регуляції глюкозного балансу беруть участь і інші гормони та системи. Печінка виконує роль буфера глюкози: поглинає надлишок глюкози з крові після їжі (під дією інсуліну) і вивільняє глюкозу в кров у міжприйомні періоди. Гіпоталамус та вегетативна нервова система реагують на зниження глюкози: активація симпатичної нервової системи та викид адреналіну стимулюють підвищення глікемії, щоб запобігти небезпечній гіпоглікемії. При тривалішому голодуванні підключаються контррегуляторні гормони – кортизол та гормон росту, які зменшують утилізацію глюкози тканинами і збільшують розщеплення жирів для забезпечення енергетичних потреб [8]. Таким чином, організм має багаторівневу систему підтримання гомеостазу глюкози.

У здоровому стані цей баланс дозволяє уникнути як надмірної гіперглікемії (яка пошкоджує судини та органи), так і гіпоглікемії (яка небезпечна для життя, особливо для мозку, що споживає глюкозу як основне пальне). Мозок людини дуже чутливий до рівня цукру в крові: при падінні глікемії нижче ~3 ммоль/л виникають симптоми нейроглікопенії (сплутаність

свідомості, судоми, кома). Тому гормональна контррегуляція (передусім викид глюкагону і адреналіну) спрацьовує при гіпоглікемії швидко, підвищуючи глюкозу і захищаючи мозок від енергодефіциту [1].

При ЦД1 описана система гомеостатичного контролю глюкози порушена внаслідок відсутності ендогенного інсуліну. Глюкагонова реакція на гіпоглікемію також може бути ослабленою при тривалому діабеті [8]. Це призводить до того, що рівень глюкози значно коливається залежно від зовнішнього надходження інсуліну та вуглеводів. Заміщення інсуліну екзогенними ін’єкціями ніколи не відтворює повністю тонкого фізіологічного регулювання. У здорової людини інсулін секретується безперервно низьким базальним рівнем та швидкими сплесками у відповідь на їжу. Сучасні методи інсулінотерапії намагаються імітувати ці патерни шляхом призначення комбінованих режимів базального та болюсного інсуліну [1]. Однак підтримання балансу глюкози при ЦД1 залишається викликом, що обумовлює необхідність удосконалення методів терапії і розробки автоматизованих систем контролю глікемії [17].

1.3 Клінічні особливості та наслідки цукрового діабету 1 типу

Діабет першого типу зазвичай маніфестує в дитячому або підлітковому віці гостро, з класичною симптоматикою: поліурія (рясне сечовипускання), полідипсія (невтамовна спрага), поліфагія (підвищений апетит) при одночасному схудненні, слабкість, зниження працездатності. Ці симптоми зумовлені гіперглікемією та відсутністю інсуліну: надлишок глюкози не засвоюється тканинами і виводиться з сечею (разом із водою, викликаючи дегідратацію і спрагу), клітини відчувають «енергетичний голод», що стимулює апетит, але без інсуліну глюкоза не може потрапити в клітини і організм вимушено розщеплює жири і білки, що призводить до схуднення. Нерідко перші прояви ЦД1 – це розвиток діабетичного кетоацидозу (ДКА): накопичення кетонових тіл і ацидоз внаслідок інсулінового дефіциту. ДКА

проявляється нудотою, блюванням, зневодненням, глибоким частим диханням (Куссмауля), запахом ацетону з рота, порушенням свідомості. Без невідкладної медичної допомоги (інсулін, регідратація, корекція електролітів) ДКА може прогресувати до діабетичної коми і смерті [21]. Таким чином, остра клінічна особливість ЦД1 – абсолютна залежність від екзогенного інсуліну: кожна людина з ЦД1 мусить щоденно отримувати інсулін, інакше ДКА розвинеться протягом лічених діб [12].

Після встановлення діагнозу і початку інсулінотерапії настає період компенсації діабету. Спочатку може спостерігатися фаза «медового місяця» – тимчасове зниження потреби в інсуліні через часткове відновлення секреції (це нестійкий стан, що триває від кількох тижнів до місяців) [25]. Надалі хворий потребує пожиттєвої інсулінотерапії. Стандартом є інтенсивний інсуліновий режим: ін’єкції базального інсуліну (1–2 рази на день) і болюсного інсуліну перед кожним прийомом їжі, або використання інсулінової помпи для безперервної підшкірної інфузії інсуліну. Дози інсуліну добираються індивідуально залежно від маси тіла, чутливості до інсуліну, вмісту вуглеводів у їжі та рівня фізичної активності. Орієнтовна початкова потреба – ~0,5 од інсуліну на 1 кг маси тіла на добу, з якої ~50% покриває базальну потребу, а решта дається у вигляді болюсів під час їжі [21]. Пацієнта навчають методам самоконтролю: вимірюванню глюкози крові глюкометром (звичайно 4 і більше разів на день) або користуванню системами безперервного моніторингу глюкози, розрахунку вуглеводних обмінних одиниць і корекції доз інсуліну [17]. Необхідні також регулярні огляди для моніторингу ускладнень (перевірка очного дна, аналіз сечі на альбумінурію, огляд стоп, оцінка неврологічного статусу тощо) [12].

Незважаючи на сучасні підходи до лікування, клінічне ведення ЦД1 залишається складним. Пацієнти повинні щоденно балансувати між ризиками гіперглікемії і гіпоглікемії. Гіперглікемія при недостатній дозі інсуліну призводить до симптомів (спраги, поліурії, стомлюваності) і у довгостроковій перспективі – до хронічних ускладнень, як згадано в підрозділі 1.1.

Гіпоглікемія при передозуванні інсуліну проявляється тремтінням, пітливістю, серцебиттям, відчуттям голоду, а при значному зниженні цукру – сплутаністю свідомості, судомами, можлива гіпоглікемічна кома. Тяжка гіпоглікемія (що потребує сторонньої допомоги) особливо небезпечна, оскільки може спричинити ураження мозку або смерть. Пацієнти з часом можуть втрачати відчуття ранніх симптомів гіпоглікемії (стан нечутливості до гіпоглікемії), що підвищує ризик раптових тяжких реакцій [17].

Таким чином, клінічне завдання при ЦД1 – підтримувати глікемію настільки близько до норми, наскільки це безпечно, уникаючи при цьому важких гіпоглікемій. Для оцінки контролю використовується показник глікованого гемоглобіну (HbA1c), що відображає середню глікемію за ~3 місяці. Як зазначалося, цільовий рівень HbA1c <7% у дорослих асоціюється зі значним зниженням ризику мікросудинних ускладнень [21]. Ще одним сучасним параметром є Time in Range (TIR) – відсоток часу, коли рівень глюкози знаходиться в цільовому діапазоні (зазвичай 3,9–10 ммоль/л) [17]. Рекомендовано досягати TIR понад 70% часу доби, що еквівалентно гарному контролю разом з мінімальним часом у гіпоглікемії <4%. Досягнення таких показників вимагає точного дозування інсуліну, своєчасної корекції, а це важко реалізувати тільки силами самого пацієнта.

У практиці лікування ЦД1 за останні роки з’явилися нові технології, що значно полегшують контроль хвороби. Це, зокрема, системи безперервного моніторування глюкози (CGM), які дозволяють відстежувати рівень глюкози в режимі реального часу, та інсулінові помпи для безперервної підшкірної доставки інсуліну. Використання цих технологій пов’язане з кращими глікемічними результатами – зниженням показників HbA1c, меншою варіабельністю глюкози та скороченням епізодів гіпоглікемії [17]. Однак і помпа, і сенсор поодинці не вирішують проблему повністю, адже пацієнт все одно має приймати багато рішень (коли і скільки інсуліну ввести, як коригувати дозу залежно від показань сенсора тощо). Тому актуальним напрямком є створення замкнених систем контролю глікемії, в яких би ввод

інсуліну автоматично регулювався на основі даних сенсора – тобто фактично йдеться про штучну підшлункову залозу [24]. Перед тим як розглянути принципи таких систем, необхідно ознайомитися з математичними моделями системи «глюкоза – інсулін», що лежать в основі алгоритмів автоматичного керування.

1.4 Сучасні підходи до моделювання системи «глюкоза – інсулін»

Для розробки ефективних алгоритмів контролю глікемії при ЦД1 науковці широко використовують математичне моделювання фізіологічної системи «глюкоза – інсулін». Моделі дозволяють імітувати поведінку рівня глюкози крові під впливом різних факторів – прийому їжі, введення інсуліну, фізичного навантаження – і таким чином тестувати стратегії інсулінотерапії віртуально, без ризику для пацієнтів. Створення адекватної моделі є складним завданням, оскільки в регуляції глікемії задіяно багато процесів (всмоктування глюкози в кишечнику, секреція гормонів, утилізація глюкози тканинами, продукція глюкози печінкою тощо). Проте за останні десятиліття було розроблено низку моделей різної складності – від спрощених «мінімальних» моделей до комплексних симуляторів організму людини.

Однією з перших стала мінімальна модель глюкозо-інсулінової динаміки, запропонована Бергманом і співавт. у 1979 р. Вона описує взаємодію інсуліну і глюкози кількома диференційними рівняннями та дозволила визначити такі важливі параметри, як інсулінова чутливість і толерантність до глюкози. Мінімальна модель стала основою для тесту толерантності до глюкози та оцінки β-клітинної функції, однак вона була розроблена на основі внутрішньовенного введення глюкози і мала обмеження у відображенні післяхарчової динаміки глікемії. Надалі з’явилися більш докладні фізіологічні моделі. Зокрема, модель Соренсена (1985) – багатокомпартментна модель, що складається з 19 диференційних рівнянь і детально описує розподіл глюкози та інсуліну між різними органами (печінка,

підшлункова залоза, периферичні тканини тощо), враховуючи також вплив глюкагону [25]. Це був важливий крок уперед, оскільки модель Соренсена дозволила симулювати як нормальний стан, так і стан діабету, однак вона виявилась дуже складною для практичного використання і не повністю відображала крайню варіабельність гіперглікемії при ЦД1.

У 1990-ті роки команда під керівництвом Кобеллі розробила низку моделей проміжної складності. Зокрема, модель Кобеллі (1998) – двокомпартментна модель (розширена мінімальна), що включала три підсистеми та враховувала вплив глюкагону [5]. Надалі ця робота продовжилась у створенні так званого UVA/Padova симулятора – найбільш просунутого інструменту моделювання ЦД1. Модель Ховорки (2002), розроблена в Кембриджі, стала одним із перших прикладів успішного використання математичного опису для керування штучною підшлунковою залозою; вона представляє нелінійну фармакокінетичну модель взаємозв'язку між підшкірним введенням інсуліну і зміною рівня глюкози крові [8]. Модель Ховорки показала високу точність у передбаченні динаміки глікемії і була використана у численних дослідженнях алгоритмів штучного підшлунка. Її недоліком є неврахування деяких контррегуляторних факторів (напр. впливу адреналіну, фізичного навантаження), але загалом вона стала "золотим стандартом" для численних симуляцій.

Особливої уваги заслуговує симулятор UVA/Padova, створений колаборацією вчених Університету Вірджинії (США) та Університету Падуї (Італія). Цей симулятор базується на моделі Далла Мана та Кобеллі (2006–2008), яка детально описує процеси травлення та засвоєння вуглеводів, динаміку інсуліну і глюкози при прийомі їжі. UVA/Padova симулятор включає популяцію віртуальних "пацієнтів" (моделі різного віку з різними параметрами), що дає змогу тестувати алгоритми контролю глікемії на широкому діапазоні сценаріїв. У 2008 році Управління з контролю за харчовими продуктами і ліками США (FDA) офіційно визнало симулятор UVA/Padova як заміну для досліджень на тваринах при преклінічному

випробуванні систем штучної підшлункової залози. Це було безпрецедентним кроком, що підкреслив довіру до адекватності моделі. Відтоді UVA/Padova використовується як стандартний інструмент для оцінки нових стратегій інсулінотерапії in silico (у комп'ютерних експериментах) перед виходом на клінічні дослідження [5].

Сучасні підходи до моделювання глюкозо-інсулінової системи можна розділити на три категорії: фізіологічні моделі, дані-орієнтовані (емпіричні) моделі та гібридні моделі [25]. Фізіологічні моделі (як описано вище) ґрунтуються на рівняннях, що відображають відомі біохімічні й фізіологічні закономірності. Моделі на основі даних використовують підходи машинного навчання, нейронні мережі та інші статистичні методи для прогнозування рівня глюкози на підставі великих масивів емпіричних даних (історії вимірювань CGM, дози інсуліну, інформація про прийом їжі тощо). Такі підходи набули популярності останнім часом, оскільки дозволяють персоналізувати прогноз глікемії для конкретного пацієнта. Гібридні моделі поєднують елементи першого і другого підходів – тобто включають базову фізіологічну структуру, але параметри уточнюються або доповнюються на основі реальних даних пацієнта. У дослідженні 2023 року було проаналізовано 138 моделей прогнозування глюкози: з них ~14% були суто фізіологічними, 54% – дані-орієнтованими, а 33% – гібридними [17]. Така класифікація підкреслює, що для створення оптимальної моделі використовуються як знання про біологічні процеси, так і потужність сучасних алгоритмів аналізу даних.

Математичні моделі слугують основою для розробки систем автоматичного керування рівнем глюкози. При проєктуванні таких систем (штучної підшлункової залози) інженери використовують модель для прогнозування майбутньої глікемії і відповідного розрахунку дози інсуліну, що повинна нормалізувати рівень цукру. Надійність і точність моделі є критичною: помилка в прогнозі може призвести до надмірного введення інсуліну (і гіпоглікемії) або до його нестачі (і гіперглікемії). Тому моделі

постійно вдосконалюються – додаються нові підсистеми (наприклад, врахування всмоктування інсуліну з підшкірної клітковини, вплив фізичних вправ, добові ритми тощо) [20]. Важливо зазначити, що навіть найкраща модель є спрощенням реальності: наприклад, моделі зазвичай не повністю враховують вплив стресу чи гормональних змін, що можуть істотно впливати на глікемію [25]. Тим не менш, сучасні моделі вже достатньо добре описують основні процеси глюкозного обміну і успішно використовуються у розвитку автоматизованих систем інсулінотерапії [17].

1.5 Автоматизовані системи інсулінотерапії (AID-системи)

Автоматизована система подачі інсуліну (Automated Insulin Delivery, AID) – це комплекс із носимого медичного пристрою та програмного забезпечення, який автоматично регулює введення інсуліну з метою підтримання цільового рівня глюкози крові. По суті, AID-система являє собою прототип штучної підшлункової залози, що відтворює функцію β-клітин по секреції інсуліну в потрібній кількості [17]. Типова сучасна AID-система включає три основні компоненти:

- **Безперервний глюкозний сенсор (CGM)** – датчик, розташований підшкірно, який кожні 5–10 хвилин вимірює рівень глюкози в інтерстиціальній рідині та передає ці дані в реальному часі.
- **Інсулінова помпа** – пристрій для підшкірної інфузії інсуліну (через катетер), що може дозовано вводити мікропорції інсуліну за командою контролера. Помпа забезпечує як базальне надходження інсуліну (постійне введення малих доз), так і болюсне (короткочасне введення більшої дози для покриття їжі або корекції гіперглікемії).
- **Контролер (алгоритм керування)** – мікропроцесор або смартфон з програмою, який отримує дані від сенсора та обчислює необхідну дозу

інсуліну, що має бути введена помпою, залежно від поточного і прогнозованого рівня глюкози.

Ці компоненти поєднані між собою за допомогою бездротового зв'язку або інтегровані в один прилад. Робота AID-системи побудована за принципом замкненого контуру: сенсор → контролер → помпа → пацієнт → сенсор. Така система постійно «відстежує» глікемію і коригує подачу інсуліну, створюючи динамічний зворотний зв'язок, аналогічний до біологічного.

Існують різні реалізації AID-систем. Більшість комерційно доступних на сьогодні – це так звані гібридні закриті системи, у яких автоматично регулюється лише базальна подача інсуліну, а пацієнт самостійно вводить болюсні дози на їжу (тобто повідомляє системі про вживання вуглеводів) [19]. Повністю закриті системи, які б не потребували участі користувача (в тому числі для болюсів), поки що знаходяться на стадії клінічних випробувань. Крім того, більшість наявних систем розраховані на використання лише інсуліну (одногормональні системи). Окремо ведуться розробки двогормональних систем, які окрім інсуліну можуть вводити ще й глюкагон або його аналог для швидкого купірування гіпоглікемії [8]. Додавання глюкагону теоретично дозволяє наблизити штучну підшлункову залозу до фізіологічної (як α-клітини підшлункової залози, що підвищують глюкозу), однак такі системи складніші (потребують двох картриджів і катетерів, інсулін і глюкагон) та дорожчі, а стабільні препарати глюкагону для помп стали доступними відносно недавно.

Перші покоління AID-систем були схвалені до використання у 2016–2018 роках (зокрема, система Medtronic 670G – перша гібридна система, дозволена FDA в 2016 р.) [38]. Нині існує вже кілька комерційних AID-рішень (Medtronic 670G/770G/780G, Tandem Control-IQ, Insulet Omnipod 5 тощо), а також відкриті (open-source) AID-системи, створені спільнотою пацієнтів і розробників (OpenAPS, AndroidAPS, Loop). Клінічні дослідження підтверджують суттєві переваги використання AID-систем порівняно з традиційними режимами: підвищується середній час у цільовому діапазоні

глікемії, знижується рівень HbA1c, скорочується кількість гіпоглікемій. Наприклад, у недавньому рандомізованому дослідженні відкритої AID-системи (алгоритм OpenAPS) пацієнти, які користувалися замкненим контуром, проводили в цільовому діапазоні на 3 години 21 хвилину більше щодоби, ніж пацієнти на стандартній сенсор-асистованій помповій терапії [24]. При цьому не було зафіксовано жодного випадку тяжкої гіпоглікемії або кетоацидозу в групі AID. Ці результати демонструють покращення глікемічного контролю і безпеки завдяки автоматизації інсулінотерапії.

Контролер AID-системи реалізує певний алгоритм керування. Найпростіший підхід – це PID-контролер (пропорційно-інтегрально-диференційний регулятор), який коригує швидкість введення інсуліну на основі поточної різниці між виміряною глюкозою і цільовим значенням, її накопиченої величини (інтеграл) та швидкості зміни глюкози (похідна) [22]. PID-алгоритми були успішно застосовані в ряді систем і мають ту перевагу, що інтуїтивно зрозумілі в налаштуванні параметрів. Більш просунуті підходи включають прогнозуючі алгоритми (MPC – Model Predictive Control), які використовують математичну модель для прогнозу майбутніх рівнів глюкози та оптимізації подачі інсуліну на декілька кроків вперед. Такі алгоритми можуть враховувати оголошені прийоми їжі, фізичну активність та інші фактори. Наприклад, відома серия досліджень Кембриджського університету застосувала MPC-контролер на базі моделі Ховорки з успішними результатами покращення нічного контролю глюкози у дітей [20]. Інший напрям – адаптивні алгоритми, що здатні навчатися на даних конкретного користувача і підлаштовувати параметри моделі під індивідуальні особливості (такі алгоритми реалізовані в деяких open-source AID) [15]. Також розробляються мультигормональні контролери для двогормональних систем, алгоритми відслідковування фізичних вправ, аварійні сигнали тощо.

Автоматизовані системи інсулінотерапії вже сьогодні демонструють революційні можливості у покращенні життя пацієнтів з ЦД1. Вони знімають частину рутинного навантаження з хворого, забезпечують більш стабільний

контроль глікемії та потенційно знижують ризик ускладнень у довгостроковій перспективі. За даними мереж спостережень, користувачі AID-систем мають вищий відсоток часу в нормоглікемічному діапазоні і нижчий рівень HbA1c порівняно з тими, хто на традиційному помповому лікуванні. Окрім медичних показників, важливим є й підвищення якості життя – пацієнти відзначають зменшення тривожності щодо нічних гіпоглікемій, більш спокійний сон, гнучкіший режим харчування і фізичної активності завдяки «розумній» помпі, яка сама приймає багато рішень.

Втім, сучасні AID-системи мають і свої обмеження. По-перше, вони все ще потребують калібрування та втручань: навіть найновіші гібридні системи вимагають від пацієнта рахувати вуглеводи та повідомляти систему про прийоми їжі (хоча деякі алгоритми можуть частково розпізнавати несподівані підйоми глюкози і самі коригувати базал або давати автоматичні болюси). По-друге, затримка дії підшкірного інсуліну (≈ 30–60 хв до піку дії) означає, що системі важко повністю запобігти післяхарчовій гіперглікемії – це стимулює дослідження альтернативних шляхів введення (інтрапервозальний, інгаляційний інсулін) та використання більш швидкодіючих аналогів інсуліну [17]. По-третє, питання безпеки і надійності: збій сенсора або помпи, помилка алгоритму чи проблеми з підключенням можуть спричинити небезпечні стани. Саме тому AID-системи оснащуються багаторівневими захисними механізмами (сигналами тривоги, автоматичним відключенням при підозрі на несправність, резервними ін’єкційними ручками для пацієнта тощо) [13]. І останнє – вартість: такі системи залишаються дорогими, що обмежує їх доступність, особливо в країнах зі страховою медициною, де не всі страхові покривають «штучну підшлункову».

Попри зазначені виклики, розвиток AID-систем стрімко триває. Перспективи включають повну автоматизацію (системи без оголошення вуглеводів), інтеграцію з моніторингом фізіологічних параметрів (наприклад, носимі фітнес-датчики для врахування фізичного навантаження), використання двох гормонів (інсуліну та глюкагону) для ще стабільнішого

контролю. Також очікується поява нових алгоритмів на основі штучного інтелекту, що ще краще передбачатимуть індивідуальні глікемічні відповіді [9]. Автоматизована інсулінотерапія вже зарекомендувала себе як ефективний інструмент для покращення контролю ЦД1: її впровадження асоціюється зі зниженням показника HbA1c в середньому на 0,5–1%, підвищенням TIR на 10–15% та зменшенням часу в гіпоглікемії. Це, в свою чергу, повинно трансльовуватися у зниження ризику довгострокових ускладнень і смертності від судинних подій у пацієнтів з ЦД1.

Науково-медичні основи автоматизованої інсулінотерапії поєднують глибоке розуміння фізіології глюкозного гомеостазу, сучасні математичні моделі та інженерні рішення. Вже сьогодні AID-системи знаменують нову еру в лікуванні ЦД1, наближаючи нас до мети – максимально нормалізувати життя людей з діабетом, мінімізувавши їхні ризики і щоденне навантаження хворобою [17].

Висновок з розділу 1

У першому розділі узагальнено науково-медичні передумови створення програмного застосунку для симуляції регуляції глюкози при цукровому діабеті 1 типу. Показано, що зростання поширеності ЦД1, висока частка гострих (гіпо- та гіперглікемічних) ускладнень і кумулятивний ризик судинних катастроф роблять підтримання HbA1c < 7 % клінічно критичним завданням. Розглянуто фізіологію гомеостазу глюкози у здорової людини та патофізіологію ЦД1, що характеризується абсолютним дефіцитом ендогенного інсуліну й нестабільними коливаннями глюкози. Огляд класичних (мінімальна модель Берґмана) і комплексних (UVA/Padova) математичних моделей показав, що вони адекватно відтворюють динаміку «глюкоза-інсулін», однак різняться складністю, вимогами до даних та обчислювальними ресурсами. Аналіз сучасних автоматизованих систем введення інсуліну (AID) засвідчив ефективність PID-/MPC-керування та

безперервного моніторингу глюкози, але наголосив на потребі у відкритих, прозорих симуляторах для валідації алгоритмів.

Таким чином, теоретичний огляд обґрунтовує вибір спрощеної, але фізіологічно інтерпретованої моделі Берґмана, інтегрованої з PD-контролером, як оптимальної основи для навчального та дослідницького симулятора. Такий підхід забезпечує баланс між біологічною достовірністю, обчислювальною швидкістю та зрозумілістю алгоритмів, що дозволяє подальші експерименти з параметрами пацієнтів, сценаріями прийому їжі й налаштуваннями контролера у наступних розділах роботи.

РОЗДІЛ 2
ОГЛЯД СУЧАСНИХ МОДЕЛЕЙ ТА ПРОГРАМНИХ ЗАСОБІВ МОДЕЛЮВАННЯ РЕГУЛЯЦІЇ ГЛЮКОЗИ ПРИ ЦД 1 ТИПУ

У цьому розділі розглянуто сучасні програмні рішення для моделювання регуляції глюкози в крові при діабеті 1 типу та автоматизованого введення інсуліну. Зокрема, буде проаналізовано існуючі симулятори та системи: UVA/Padova (університетський симулятор діабету), його відкриту реалізацію simglucose, а також відкриті проекти замкненого контуру OpenAPS і Loop, і комерційну платформу Medtronic CareLink. Буде показано різні підходи до моделювання глікемії – від диференційних математичних моделей до емпіричних алгоритмів, а також здійснено класифікацію цих моделей (наприклад, на основі рівнянь (ODE) чи із застосуванням методів машинного навчання). Такий огляд дозволить виявити сильні та слабкі сторони існуючих підходів та обґрунтувати вибір власного рішення.

2.1 Огляд існуючих симуляторів та ПЗ для моделювання діабету 1 типу

Одним з найвідоміших інструментів для досліджень діабету 1 типу є симулятор, розроблений Університетом Вірджинії (UVA) спільно з Університетом Падуї – **UVA/Padova Type 1 Diabetes Simulator**. Він реалізує складну фізіологічну модель глюкозо-інсулінової динаміки на основі систем диференційних рівнянь, що охоплюють процеси метаболізму глюкози, всмоктування вуглеводів, фармакокінетику інсуліну тощо. Симулятор має популяцію з 300 віртуальних пацієнтів (по 100 дітей, підлітків та дорослих) з діабетом 1 типу. Кожен віртуальний «пацієнт» характеризується унікальним набором параметрів моделі (чутливість до інсуліну, швидкість абсорбції, інсулінова доза тощо), що дозволяє відтворювати індивідуальні профілі.

Вбудованого контролера в UVA/Padova симуляторі немає – його призначено для оцінки алгоритмів керування (наприклад, штучної підшлункової) шляхом моделювання їх роботи in silico. Важливо, що цей симулятор став першим, затвердженим FDA як заміна тваринних моделей для преклінічних випробувань стратегій інсулінотерапії (у 2008 році) [5]. Формат розповсюдження симулятора – дослідницьке програмне забезпечення (початково закрите, доступне за домовленістю для академічних цілей). Модель – детермінована (ODE). Контролер: відсутній (передбачено зовнішнє підключення).

Simglucose – це відкритий Python-симулятор діабету, що реалізує модель UVA/Padova (версії 2008 року) для потреб машинного навчання та контролю. По суті, він є портом згаданого FDA-затвердженого симулятора на мову Python з дотриманням його математичних рівнянь. Simglucose включає 30 віртуальних пацієнтів (10 дорослих, 10 підлітків, 10 дітей), які статистично репрезентують більш велику популяцію. Модель симулятора – розподілена система ODE, успадкована від UVA/Padova, що дозволяє розраховувати рівень глюкози в крові на основі введеного інсуліну та спожитих вуглеводів у часі. Контролер як такий до складу simglucose не входить (за замовчуванням симулятор просто моделює глікемію при заданому профілі інсулінотерапії), однак він сумісний із алгоритмами навчання з підкріплення (RL) та іншими контролерами, оскільки має API у форматі OpenAI Gym. Simglucose розповсюджується як open-source (ліцензія MIT) і доступний на GitHub [33]. Завдяки відкритості та крос-платформеності (Python) цей симулятор широко використовується для експериментів з автоматизованим керуванням глікемією.

OpenAPS (Open Artificial Pancreas System) – це відкрита DIY-система штучної підшлункової залози, орієнтована на практичне використання хворими на діабет 1 типу. На відміну від наведених вище симуляторів, OpenAPS не моделює віртуального пацієнта, а працює з реальними даними користувача (безперервний моніторинг глюкози, інсулінова помпа) у режимі

реального часу. Система складається зі спеціально налаштованого мікрокомп'ютера (наприклад, Raspberry Pi) або іншого пристрою, підключеного до CGM-сенсора та інсулінової помпи, та виконує замкнений алгоритм контролю. Алгоритм OpenAPS автоматично коригує подачу інсуліну (переважно шляхом тимчасових базальних доз) кожні 5 хвилин на основі поточних показників глюкози та прогнозу їх зміни. Цей алгоритм відомий як референсний дизайн oref0/oref1 і є по суті пропорційно-інтегральним регулятором зі врахуванням «інсуліну на борту» (Insulin on Board) – тобто кількості активного інсуліну, що вже введений але ще діє [24]. Модель пацієнта як така явно не використовується – система спирається на емпіричні формули й налаштування, задані користувачем (чутливість до інсуліну, вуглеводний коефіцієнт тощо). Графічний інтерфейс OpenAPS не має штатного (конфігурація здійснюється через командний рядок), однак для візуалізації даних зазвичай використовується сумісний веб-застосунок Nightscout. Проект OpenAPS є повністю відкритим (відкритий вихідний код контролера, документація тощо)і поширюється спільнотою ентузіастів безкоштовно. Така система вперше дозволила пацієнтам, «не чекаючи» комерційних рішень, самим зібрати собі прототип штучної підшлункової залози, сумісний із наявними пристроями.

Loop – ще одна відкрита AID-система (Automated Insulin Delivery), розроблена спільнотою, яка працює на пристроях Apple (iPhone, iPod) і забезпечує зручний мобільний інтерфейс користувача. На відміну від OpenAPS, Loop реалізовано як додаток з графічним інтерфейсом, що взаємодіє з сумісною інсуліновою помпою (наприклад, старі моделі Medtronic або Omnipod) через Bluetooth/Radio-адаптер та отримує дані з CGM. Алгоритм Loop побудований на основі принципів model predictive control (MPC): кожні 5 хвилин система будує короткостроковий прогноз рівня глюкози, виходячи з поточного стану (глюкози, активного інсуліну, внесених даних про вуглеводи), і підбирає таке коригування дози, щоб прогнозовані значення глюкози наблизились до цільового [15]. Для цього Loop використовує

спрощену модель обчислення майбутнього рівня глікемії, що враховує фармакодинаміку швидкодіючого інсуліну (типово тривалість дії ~6 год) та абсорбцію вуглеводів, закладені через налаштування користувача. Контролер Loop може автоматично коригувати як базальну швидкість інсуліну, так і рекомендувати болюси, щоб попередити значні відхилення глюкози. Модель пацієнта: емпірична (MPC-модель, основана на параметрах, заданих користувачем). GUI: повнофункціональний мобільний додаток (iOS). Ліцензія проекту – відкрита (MIT), однак для використання необхідно самостійно зібрати додаток з вихідних кодів (через вимоги регуляторики додаток не розповсюджується в App Store офіційно). Проект Loop надає більшу «дружність» до користувача та гнучкість у порівнянні з OpenAPS, і наразі (станом на 2025 р.) використовується багатьма технічно обізнаними пацієнтами, хоча й не є офіційно схваленим методом терапії.

Medtronic CareLink – це фірмове веб-програмне забезпечення компанії Medtronic для моніторингу та аналізу даних терапії діабету. На відміну від попередніх систем, CareLink не є симулятором чи контролером, а виконує допоміжну функцію збирання даних з пристроїв (інсулінових помп, CGM, глюкометрів) та генерації звітів для пацієнта і лікаря. Дані з помпи (в тому числі в рамках роботи гібридної замкненої системи MiniMed) передаються через інтернет у хмарний сервіс CareLink Personal, де користувач може переглядати графіки рівня глюкози, дози інсуліну, тенденції та події. Лікарі можуть використовувати CareLink Pro/Clinic для віддаленого спостереження за пацієнтом. Таким чином, CareLink слугує інструментом підтримки прийняття рішень і оптимізації терапії, але не здійснює автоматичного керування глікемією самостійно (контроль реалізовано у самій помпі). У випадку сучасних інсулінових помп Medtronic (серії 670G, 770G, 780G), які працюють в режимі авто-базалу (Auto Mode), CareLink відображає результати роботи внутрішнього алгоритму помпи [19]. Відомо, що контролер Medtronic MiniMed 670G побудований на класичному PID-алгоритмі (пропорційно-інтегрально-диференційному регуляторі) з модулем зворотного зв’язку по

інсуліну [17]. Цей алгоритм кожні 5 хвилин розраховує мікроболюси інсуліну, щоб утримувати глюкозу поблизу цільового значення ~120 мг/дл, враховуючи актуальний сенсорний рівень, тенденцію зміни та залишок активного інсуліну. Інтерфейс CareLink – веб-додаток із захищеним доступом; користувачі отримують графічні звіти, діаграми тенденцій, статистики тощо. ПЗ CareLink є безкоштовним для власників пристроїв Medtronic, але закритим (пропрієтарним) – вихідний код не розголошується, використання можливе лише з сумісними пристроями компанії.

2.2 Порівняльна характеристика функціоналу

Для наочного співставлення можливостей розглянутих рішень, наведемо їх основні характеристики у вигляді таблиці (табл. 2.1). Включено такі аспекти: категорія (симулятор чи реальна система), тип моделі або алгоритму, наявність і тип контролера, інтерфейс для користувача, формат розповсюдження (ліцензія, відкритість), а також підтримка різних профілів пацієнтів.

Таблиця 2.1

Порівняння існуючих симуляторів та AID-систем для діабету 1 типу

Платформа	Категорія / призначення	Модель / профілі	Алгоритм контролю	Інтерфейс користувача	Ліцензія / доступність
UVA/Padova Simulator	Науковий симулятор глікемії (in silico випробування стратегій)	Детальна фізіологічна ODE-модель; 300 віртуальних пацієнтів (100 дитячих, 100 підліткових, 100 дорослих)	Вбудований контролер відсутній (підтримка зовнішніх контролерів для тестування)	Текстовий/файловий (відсутній GUI; інтерфейс через код або спеціалізовані утиліти)	Пропрієтарне ПЗ для дослідників (FDA-визнаний симулятор, доступ за угодою)

Продовження таблиці 2.1

Платформа	Категорія / призначення	Модель / профілі	Алгоритм контролю	Інтерфейс користувача	Ліцензія / доступність
Simglucose	Відкритий симулятор для досліджень (RL, моделювання)	ODE-модель (UVA/Padova 2008); 30 віртуальних профілів (дорослий, підліток, дитячий)	Вбудований контролер відсутній; підтримка підключення будь-яких алгоритмів (через OpenAI Gym API)	Мінімальний (консольний запуск, графіки генеруються скриптами)	Open-source (MIT ліцензія) [2]; Python-пакет вільно доступний
OpenAPS	DIY-система автоматичного введення інсуліну (реальний користувач)	Явна модель пацієнта не використовується (реальні дані CGM; профіль користувача: інсулін-чутливість, коеф. вуглеводів тощо)	Так, замкнений контролер (алгоритм oref0/oref1: модифікований PID з прогнозом на основі IOB) – автоматичне коригування базалу кожні 5 хв	Немає власного GUI (налаштування через Shell); для моніторингу застосовують веб-візуалізацію (Nightscout)	Open-source (вихідний код і документація у відкритому доступі); потребує сумісних пристроїв (CGM + помпа)
Loop	DIY-система автоматичного введення інсуліну (реальний користувач)	Спрощена модель прогнозування глікемії (MPC) із параметрами під користувача; профіль: налаштування інсуліну/вуглеводів, тривалість інсуліну тощо	Так, замкнений контролер (алгоритм на основі MPC: прогноз + корекція болюсів/базалу кожні 5 хв)	Повноцінний мобільний додаток (iOS) з графічним інтерфейсом (моніторинг, керування болюсами, режимами)	Open-source (MIT); доступний для самостійної збірки; офіційно не сертифікований (експериментальне використання)
Medtronic CareLink	Комерційне веб-ПЗ для моніторингу терапії (реальні дані)	Модель як така відсутня (реальні пацієнти); підтримує будь-які профілі користувачів з пристроями Medtronic (через завантаження даних)	Контролер в помпі (наприклад, в MiniMed 670G: PID-алгоритм з інсуліновим зворотнім зв’язком; в CareLink лише відображення його роботи)	Веб-інтерфейс (онлайн-акаунт) для перегляду звітів, графіків, статистики; інтеграція з мобільним додатком для автозавантаження	Пропрієтарне ПЗ (Medtronic); безкоштовне для користувачів пристроїв компанії; закритий код, використання лише з сумісними помпами/CGM

Як видно з таблиці, існуючі рішення можна умовно поділити на дві категорії: імітатори (симулятори) та реальні системи автоматичного контролю (APS). До перших належать UVA/Padova та його відкриті аналоги (simglucose), які забезпечують детальне моделювання фізіології та варіативності пацієнтів, проте не мають інтегрованого контролера і потребують зовнішнього алгоритму для керування подачею інсуліну [33]. Їх сильна сторона – висока

точність і валідація (UVA симулятор визнаний регулятором), можливість безпечного випробування стратегій на великій вибірці in silico [5]. Натомість слабкими сторонами є складність і закритість: оригінальний UVA/Padova симулятор недоступний широкому загалу, а його модель містить багато параметрів, що ускладнює використання без глибокої експертизи. Open-source версія (simglucose) розв'язує проблему доступності та прозорості, однак все ще вимагає від користувача навичок програмування (відсутній зручний GUI) і знань для налаштування експериментів [33].

Другу категорію становлять замкнені системи керування діабетом – як відкриті (OpenAPS, Loop), так і комерційні (Medtronic та інші). Їх перевага – безпосереднє практичне застосування: вони здатні автоматично керувати дозуванням інсуліну у реального пацієнта, зменшуючи навантаження на користувача і покращуючи контроль глікемії [24]. Відкриті DIY-рішення (OpenAPS, Loop) продемонстрували ефективність у спільноті пацієнтів, даючи можливість гнучкого налаштування під індивідуальні потреби. Їх сильна сторона – інноваційність і гнучкість: алгоритми можуть швидко вдосконалюватись спільнотою, параметри піддаються тонкому налаштуванню (наприклад, користувач Loop може задати цільовий діапазон глюкози, чого не дозволяють комерційні системи) [15]. Водночас ці системи неофіційні та потребують технічних навичок для впровадження; питання безпеки і відповідальності лягають на самого користувача. Комерційні ж рішення (як-от Medtronic 670G/770G з платформою CareLink) є сертифікованими і підтримуються виробником, що гарантує певний рівень надійності і простоти для кінцевого користувача [19]. Їх контролери, як правило, відносно консервативні (PID-регулятори), що пояснюється пріоритетом безпеки та вимогами регуляторних органів. Недоліком закритих систем є менша прозорість (користувач не має доступу до алгоритму, не може його налаштувати під себе) та вища вартість [17].

Таким чином, аналіз існуючих аналогів показує, що універсального рішення не існує: наукові симулятори забезпечують глибоке розуміння

процесів, але не інтегрують зручного контролера; натомість реальні APS-системи автоматизують лікування, але або недоступні широкому загалу (через регуляторні обмеження або вартість), або вимагають компромісу між гнучкістю і простотою. Ці спостереження враховуються при виборі власного підходу в даній роботі.

2.3 Обґрунтування вибору власного підходу

Беручи до уваги результати аналізу, для подальшої роботи обрано розробку власного симулятора регуляції глюкози з інтегрованим контролером на основі спрощеної математичної моделі. Зокрема, пропонується використати адаптовану мінімальну модель Бергмана (Bergman) для системи «глюкоза-інсулін» у поєднанні з PD-регулятором (пропорційно-диференційним контролером) для автоматичного дозування інсуліну. Такий вибір зумовлений низкою факторів, що відповідають поставленим цілям дослідження:

- **Прозорість та наочність моделі**. Мінімальна модель гомеостазу глюкози (вперше запропонована Бергманом у 1979 р.) є концептуально простою та добре вивченою в науковій літературі. Вона описує динаміку глюкози в крові через невелику кількість змінних (наприклад, концентрацію глюкози та інсуліну в плазмі, а також «віддалений» ефект інсуліну) за допомогою системи звичайних диференційних рівнянь [4]. За десятиліття використання ця модель зарекомендувала себе як інструмент для оцінки чутливості до інсуліну та інших параметрів метаболізму, водночас залишаючись достатньо простою для аналітичного розуміння. Обравши адаптацію моделі Бергмана, ми отримуємо прозору структуру, де кожен параметр має фізіологічний зміст (наприклад, SI – інсулінова чутливість, SG – ефективність глюкози тощо). Це важливо для навчальної та дослідницької цінності: на основі такої моделі легше пояснити принципи регуляції глікемії, ніж на основі «чорної скриньки».

- **Простота та низькі вимоги до даних**. Мінімальна модель є менш параметризованою, ніж комплексні симулятори на кшталт UVA/Padova. Відповідно, для її налаштування не потрібні великі масиви пацієнтських даних – достатньо кількох ключових параметрів, які можна оцінити з клінічних джерел або літератури. Це робить можливим побудову універсального демонстраційного інструменту, який не прив'язаний до конкретного пацієнта. Крім того, обчислювальна складність мінімальної моделі низька, що дозволяє виконувати симуляцію в режимі, близькому до реального часу, навіть у простому середовищі (наприклад, на домашньому комп'ютері) [25]. Таким чином, обраний підхід є ресурсно ефективним і придатним для інтерактивної реалізації.
- **Інтегрування простого PD-контролера**. Для автоматичного регулювання рівня глюкози буде використано класичний підхід – пропорційно-диференційний (PD) контролер. Такий вибір обґрунтований тим, що PD(PID)-регулятори вже довели свою ефективність на практиці: перша сертифікована гібридна система (Medtronic 670G) застосувала саме PID-алгоритм для контролю базального інсуліну [19]. PD-регулятор є підмножиною PID, що не має інтегральної складової; це спрощує налаштування і усуває ризик інтегрального переналаштування (windup), що особливо важливо при обмежених даних та відсутності повноцінної адаптації [39]. У контексті мінімальної моделі PD-контролер може бути налаштований на основі інтуїтивно зрозумілих критеріїв (коефіцієнт пропорційності визначає, наскільки агресивно вводиться інсулін при відхиленні глюкози від цілі, а диференційний коефіцієнт – як система реагує на швидкі зміни глюкози) [22]. Така схема забезпечить стабільне керування: при правильному підборі коефіцієнтів можна досягти плавного виходу глюкози на цільовий рівень після порушень (наприклад, прийому їжі) з мінімальними перерегулюваннями. Хоча існують більш складні

алгоритми (MPC, адаптивні або з елементами штучного інтелекту), PD-контролер обрано з міркувань демонстраційної простоти та відповідності завданню. [20]

- **Реалізація на мові Python з GUI**. Запропоноване рішення буде реалізовано в середовищі Python, що обумовлено його популярністю у науковому співтоваристві та наявністю численних бібліотек для обчислень та побудови інтерфейсів [30]. Використання мови Python дозволяє легко інтегрувати математичну модель (через бібліотеки SciPy, NumPy для розв'язання ODE) та реалізувати контролер [31]. Крім того, засобами Python (Tkinter або інші GUI-фреймворки) буде створено простий графічний інтерфейс користувача для взаємодії з симулятором [27]. Планується забезпечити відображення основних змінних (глюкози у часі, трендів інсуліну), керування параметрами моделі та контролера через GUI, що зробить інструмент наочним і зручним у використанні. Таким чином, наше ПЗ поєднуватиме в собі функціональність симулятора і контролера: користувач зможе задати сценарії (наприклад, прийом їжі, болюс інсуліну вручну) і спостерігати, як PD-контролер адаптує подачу інсуліну для підтримання глікемії.

Отже, власний підхід – адаптована мінімальна модель з PD-контролем – було обрано завдяки його прозорості, простоті та достатній ефективності для моделювання базових принципів регуляції глюкози. Це рішення, з одного боку, запозичує ідеї з перевірених часом моделей (мінімальний модельний підхід Бергмана) та алгоритмів керування (PID, який використовується і в комерційних системах), а з іншого – забезпечує гнучкість відкритої реалізації (Python), дозволяючи надалі розширювати й модифікувати систему. Очікується, що така модельно-алгоритмічна зв'язка продемонструє основні характеристики системи штучної підшлункової залози, залишаючись інтуїтивно зрозумілою та доступною для подальших удосконалень.

Висновок з розділу 2

В ході аналізу існуючих аналогів програмного забезпечення для моделювання діабету 1 типу встановлено, що кожен з підходів має свої переваги та обмеження. Докладно розглянуто симулятори UVA/Padova та simglucose, які надають можливість ін-силіко експериментів з високою вірогідністю, проте не інтегрують готових рішень для контролю. Вивчено відкриті системи OpenAPS та Loop, що реалізують замкнений контроль глікемії у реальних користувачів і тим самим демонструють практичну здійсненність штучної підшлункової залози – водночас ці рішення або потребують значних технічних зусиль, або обмежені у поширенні. Комерційна платформа Medtronic CareLink проілюструвала, як промислові гібридні системи використовують консервативні PID-алгоритми для гарантії безпеки, надаючи користувачу інструменти моніторингу без можливості втручання в роботу контролера.

Порівняльний аналіз показав, що актуальним завданням є створення системи, яка поєднала б переваги обох категорій – тобто мала б прозору і просту модель, інтегрований автоматичний контролер та зручний інтерфейс користувача. Саме тому в даній роботі запропоновано власний підхід на базі адаптованої мінімальної моделі з PD-регулятором, реалізований у відкритому середовищі. Такий вибір обґрунтовано необхідністю отримати інструмент, придатний як для наочного дослідження процесів регуляції глюкози, так і для перевірки алгоритмів керування на спрощеній, але адекватній моделі.

РОЗДІЛ 3
ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИМУЛЯЦІЇ РЕГУЛЯЦІЇ ГЛЮКОЗИ ПРИ ЦД 1 ТИПУ

У цьому розділі описано інструментарій та технології, використані для реалізації програмного забезпечення моделювання регуляції рівня глюкози в крові при діабеті 1 типу. Як мову програмування вибрано Python 3.12.4 – одну з найбільш популярних мов розробки, що входить до топ-3 мов за поширеністю серед розробників. Python є інтерпретованою об'єктно-орієнтованою мовою високого рівня з динамічною семантикою, яка забезпечує швидку розробку та читабельність коду. Середовище розробки – Visual Studio Code – обране через зручність і широке поширення.

Для побудови графічного інтерфейсу користувача (GUI) застосовано стандартну бібліотеку Python – Tkinter, яка є штатним інтерфейсом до інструментарію Tcl/Tk і постачається разом із Python на Windows, Linux та macOS. Використано модуль tkinter.ttk з набором стильових віджетів Tk 8.5, що відокремлюють логіку відображення від поведінки елементів інтерфейсу. Візуалізація результатів моделювання здійснюється за допомогою бібліотеки Matplotlib, інтегрованої в інтерфейс через клас FigureCanvasTkAgg для відображення графіків безпосередньо в вікні програми. Графічні компоненти (траєкторії змінних моделі, управління u(t), збурення D(t)) відображаються різними кольорами і шкалуються динамічно; передбачено індикаторні лінії порогових значень глюкози для наочності. Архітектура програмного застосунку передбачає розподіл на модуль математичної моделі (обчислення диференційних рівнянь) та модуль GUI. Щоб забезпечити реактивність інтерфейсу, обчислення виконуються у фоновому потоці за допомогою модуля threading, що дозволяє уникнути блокування GUI під час чисельної симуляції (відповідний підхід рекомендований для паралельного виконання вводу/виводу в однопоточному середовищі Python). Така побудова забезпечує

гнучкість конфігурації (через зовнішні файли налаштувань у форматі YAML) та швидкість розробки, оскільки основні компоненти (алгоритми моделі, візуалізація, інтерфейс) використовують високорівневі бібліотеки з оптимізованим внутрішнім кодом. Нижче детально розглянуто вибір мови програмування, бібліотек для математичного моделювання, засобів створення інтерфейсу та вимоги до апаратного і програмного забезпечення.

3.1 Мова програмування та середовище розробки

Основою програмної реалізації є мова **Python** (версія 3.12.4). Вибір Python обґрунтовано її універсальністю та ефективністю для наукових обчислень і швидкого прототипування. Python має простий синтаксис, високу читабельність коду та підтримує кілька парадигм програмування, що знижує вартість супроводу програмного коду. Важливою перевагою є наявність великої стандартної бібліотеки та розвиненої екосистеми наукових пакетів. Інтерпретатор Python і стандартні бібліотеки доступні безкоштовно для всіх основних платформ (Windows, Linux, macOS), тому розроблене програмне забезпечення є крос-платформним [30]. Python здобув широке визнання у сфері аналізу даних, машинного навчання та біомедичних обчислень; фактично, він став однією з домінуючих мов у науково-інженерній спільноті [34]. Це підтверджується як опитуваннями розробників (Python стабільно серед лідерів за популярністю), так і наявністю спеціалізованих бібліотек для біомедичних задач (наприклад, BioPython, NumPy, SciPy тощо) [23].

Для розробки застосунку було використано інтегроване середовище **Visual Studio Code (VS Code)**. Це сучасний багатоплатформений редактор коду з відкритим вихідним кодом, що підтримує Python через розширення та надає зручні інструменти відлагодження, інтеграції з системами контролю версій та керування середовищами . Популярність VS Code серед розробників надзвичайно висока – за даними опитування 2023 року, приблизно 74% професійних розробників використовували VS Code як основне середовище

розробки [34]. Такий вибір зумовлений можливістю швидко запускати і тестувати Python-код, автодоповненням, інтегрованим терміналом та підтримкою Git, що прискорює цикл «редагування-тестування».

Python має вбудовані засоби для створення настільних GUI-додатків. Зокрема, використано бібліотеку Tkinter, яка є стандартним інтерфейсом Python до графічного тулкіту Tk. Tkinter поставляється разом з Python і вважається «де-факто» стандартним рішенням для GUI на Python. Перевагою Tkinter є його простота та інтеграція в стандартну бібліотеку – для використання не потрібна стороння інсталяція [27]. До складу Tkinter входить модуль ttk (Tk Themed Widgets), що надає набір сучасних віджетів із підтримкою тем оформлення, запроваджених у Tcl/Tk 8.5. Бібліотека ttk дозволяє створювати більш привабливий інтерфейс користувача та відокремлює логіку роботи віджетів від їхнього стилю, що покращує узгодженість дизайну застосунку [28].

Таким чином, поєднання мови Python та середовища VS Code забезпечило швидку імплементацію задуманої функціональності. Python-додаток легко запускати та відлагоджувати на різних платформах, а GUI на Tkinter не потребує додаткових залежностей. У наступних підрозділах розглянуто використані бібліотеки та інструменти більш детально, особливо з точки зору їх застосування в задачі математичного моделювання фізіологічних процесів та створення зручного інтерфейсу.

3.2 Бібліотеки та інструменти для математичного моделювання

Реалізація математичної моделі регуляції глюкози ґрунтується на використанні низки потужних бібліотек Python, які забезпечують ефективні чисельні обчислення та зручність розробки. До основного технологічного стеку увійшли такі бібліотеки: NumPy, SciPy, Matplotlib, Pandas, а також допоміжні модулі стандартної бібліотеки (logging, threading) та зовнішні

пакети для конфігурації та графіки (YAML, PIL). Нижче наведено обґрунтування використання кожної з них.

- Бібліотека **NumPy (Numerical Python)** використовується для роботи з багатовимірними масивами даних та швидкого виконання математичних операцій. NumPy є фундаментальним пакетом для наукових обчислень мовою Python. Він надає високопродуктивний N-вимірний масив ndarray та цілу низку оптимізованих рутин для операцій над масивами – алгебра, статистика, Фур'є-аналіз, генерація випадкових чисел тощо. Ключовою перевагою NumPy є те, що більшість операцій над масивами реалізовано на мовах C/Fortran і виконуються в скомпільованому коді, завдяки чому досягається швидкодія, близька до C-рівня. Наприклад, елементо-покWise операції над великими масивами виконуються NumPy значно ефективніше, ніж ітерація в чистому Python, оскільки внутрішньо працює вже скомпільований алгоритм. У сфері моделювання біологічних процесів такі можливості NumPy є критичними, адже дозволяють обробляти часові ряди даних (глюкоза, інсулін тощо) та обчислювати значення станів моделі на кожному кроці інтегрування зі значною швидкістю. NumPy також слугує базисом для інших наукових бібліотек Python – масиви ndarray є стандартом обміну даними між функціями SciPy, Pandas, Matplotlib та ін., що спрощує інтеграцію їх використання [23].
- Пакет **SciPy (Scientific Python)** – це розширення можливостей NumPy, яке містить широкий набір алгоритмів і чисельних методів, зокрема для інтегрування диференційних рівнянь, оптимізації, обробки сигналів та статистичного аналізу. SciPy є одним із ядрових компонентів наукової екосистеми Python для математики, науки і інженерії. У даному проекті основним є модуль SciPy.integrate, який містить засоби чисельного розв'язування звичайних диференційних рівнянь (ЗДР). Зокрема, використовується функція solve_ivp (solve initial value problem) для

інтегрування системи ЗДР, що описує динаміку глюкози та інсуліну. Застосування solve_ivp обґрунтовано тим, що розробники SciPy рекомендують саме її для нового коду, тоді як застаріла функція odeint вважається менш гнучкою. У документації SciPy явно зазначено: «For new code, use scipy.integrate.solve_ivp to solve a differential equation», тобто solve_ivp є сучаснішим інтерфейсом з низкою покращень. На відміну від odeint, функція solve_ivp підтримує кілька чисельних методів (наприклад, інтегратори Runge-Kutta 4(5) порядку, Radau та ін.), дозволяє задавати події (events) для відстеження певних умов під час інтегрування, а також уніфікує порядок аргументів функції правих частин (формат f(t, y) за замовчанням). Це зробило solve_ivp зручним вибором для реалізації математичної моделі, оскільки модель представляється системою ОДУ, а SciPy забезпечує надійне та перевірене рішення для її чисельного розв’язання. Додатково SciPy використовувалася для ряду інших математичних задач, як-от інтерполяція чи пошук коренів (за необхідності в інших частинах проекту), однак ключовою саме для даного розділу є функціональність інтегратора ОДУ [31].

- Для побудови графіків і візуалізації результатів моделювання використано бібліотеку **Matplotlib**. Matplotlib – це потужна бібліотека для створення статичних, анімаційних та інтерактивних візуалізацій в Python. Вона дозволяє генерувати графіки, близькі за якістю до публікаційних, у різноманітних форматах (PNG, PDF тощо) і може працювати у багатьох середовищах – скриптах, ноутбуках Jupyter, веб-додатках та різних GUI-фреймворках. Саме остання властивість критично важлива: Matplotlib підтримує декілька бекендів для відображення графіки, зокрема модуль matplotlib.backends.backend_tkagg, який забезпечує інтеграцію з Tkinter. Завдяки цьому графіки можуть відображатися безпосередньо в інтерфейсі програми через віджет Canvas. В нашому застосунку

Matplotlib використовується для побудови часових залежностей концентрації глюкози (змінні моделі x_1 та x_3) та профілів зовнішніх впливів (u(t) – доза інсуліну, D(t) – надходження глюкози). Гнучкість бібліотеки дозволила налаштувати оформлення графіків: нанесені горизонтальні лінії порогових рівнів глікемії, різні кольори кривих та легенду для розрізнення змінних. Посилання на офіційну документацію Matplotlib та приклади використання підтверджують можливість вбудовування графіків у Tkinter-додаток за допомогою класу FigureCanvasTkAgg та панелі інструментів NavigationToolbar2Tk для інтерактивних дій (масштабування, збереження графіка тощо). Таким чином, Matplotlib забезпечує високоякісну візуалізацію результатів, що є важливою складовою наочності програмного продукту [18].

- Бібліотека **Pandas** застосована для роботи з табличними даними та часовими рядами. Pandas надає структури даних на кшталт DataFrame і високорівневі засоби для маніпуляції даними (фільтрація, агрегування, злиття тощо). Ця бібліотека спеціально розроблена для аналізу даних і широко використовується у науковій та фінансовій сферах. Згідно з офіційною документацією, Pandas є відкритим BSD-ліцензованим пакетом, що забезпечує високопродуктивні та зручні у використанні структури даних для мови Python. У рамках нашого проекту Pandas спрощує завантаження і збереження даних профілів пацієнтів (наприклад, параметрів моделі або історичних даних вимірів глюкози) у форматі CSV або Excel, а також може використовуватися для обробки результатів моделювання (побудови таблиць показників, розрахунку статистик по симуляціях тощо). Хоча основна динаміка моделі обчислюється на масивах NumPy, Pandas зручний для підготовки вхідних даних та пост-обробки результатів, особливо коли йдеться про набір експериментів або сценаріїв, які треба порівняти [26].

- Модуль **logging** із стандартної бібліотеки Python використано для організації ведення журналу подій застосунку. Logging надає гнучкий механізм логування для додатків і бібліотек. У програмі симуляції контролю глюкози цей модуль дозволяє фіксувати важливі події: запуск і зупинку моделювання, виникнення нештатних ситуацій (наприклад, гіпо- чи гіперглікемії, якщо реалізована така перевірка), повідомлення про збереження результатів тощо. Використання logging замість виводу повідомлень у консоль робить програму більш професійною – всі повідомлення можуть зберігатися у файл (наприклад, app.log) з зазначенням часу і рівня важливості (INFO, WARNING, ERROR). Це полегшує відлагодження та подальший аналіз роботи програмного забезпечення. Бібліотека logging гнучко налаштовується (можна встановлювати рівні логування, формат повідомлень, різні Handler для виводу логів), що було застосовано для розмежування інформаційних повідомлень та повідомлень про помилки або виняткові ситуації [30].
- Для зберігання та завантаження конфігураційних параметрів застосунку використано формат **YAML (Yet Another Markup Language)**. YAML – це зручний для читання людиною формат серіалізації даних, який широко використовується для конфігураційних файлів. У проекті визначено файл (або файли) у форматі YAML, в якому задано основні параметри моделі та сценарії симуляції: наприклад, параметри моделі пацієнта (показники чутливості до інсуліну, швидкості абсорбції глюкози тощо), налаштування інтерфейсу (кольори графіків, межі шкал) або сценарій введення глюкози (профіль харчування) та інсуліну. Формат YAML було обрано через його читабельність – на відміну від JSON чи XML, YAML дозволяє зберігати дані у вигляді структур з відступами, наближеному до Python-словників, і підтримує коментарі. Для роботи з YAML у Python використано бібліотеку PyYAML: вона дозволяє легко завантажити дані з .yml файлу у вигляді словників Python

і так само зберегти. Таким чином, конфігурація моделі і сценаріїв можуть змінюватись без модифікації вихідного коду програми – достатньо відредагувати YAML-файл. Це підвищує гнучкість програмного забезпечення та зручність експериментування з різними умовами моделювання [3].

- Для реалізації конкурентного виконання задач (фонової обробки) використано модуль **threading** стандартної бібліотеки. Хоча Python має глобальну блокування інтерпретатора (GIL), що не дозволяє двом потокам виконувати Python-код одночасно, використання потоків все ж доцільне для асинхронного виконання задач вводу/виводу та підтримки інтерфейсу. У нашому застосунку чисельне інтегрування системи рівнянь виконується у окремому потоці, запущеному паралельно з основним потоком GUI. Це зроблено для того, щоб інтерфейс (головне вікно Tkinter) залишався чутливим до дій користувача – реагував на натискання кнопок, перемикання вкладок тощо – навіть під час інтенсивних обчислень моделі. Потік симуляції періодично повідомляє головний потік про нові значення (наприклад, додаючи точки на графіку), що реалізовано через механізми, безпечні щодо потоків (виклики методів Tkinter через queue або after). Таким чином, threading забезпечує псевдо-паралельність: у той час як обчислення моделі виконуються (можливо, у бібліотеці SciPy, яка реалізована на C і тимчасово відпускає GIL під час інтегрування), інтерфейс продовжує оновлюватись. Альтернативою могла б бути реалізація на multiprocessing або асинхронних викликах, але для поставленої задачі threading є достатнім і простішим у реалізації рішенням [30].
- Для роботи з зображеннями та графічними файлами використовується бібліотека **PIL** (в сучасній реалізації – **Pillow**). В нашому програмному забезпеченні PIL застосовано для кількох допоміжних цілей: завантаження користувацьких графічних елементів інтерфейсу

(наприклад, іконки програми або зображення для фону вікна), збереження графіків, побудованих Matplotlib, у форматах PNG/JPEG для подальшого використання у звітах, а також можливого експорту результатів симуляції у вигляді зображень. Бібліотека Pillow дозволяє відкривати, створювати, редагувати та зберігати зображення багатьох форматів засобами кількох простих викликів. Зокрема, графік Matplotlib може бути збережений через метод Figure.savefig(), але для додаткової обробки або об'єднання з іншими зображеннями (наприклад, накладення логотипу чи підпису) можна використати PIL. Також PIL могла б стати в пригоді для генерації зображень інтерфейсу (наприклад, індикаторів), але в поточній реалізації основний акцент зроблено на побудові графіків, тому PIL виконує другорядну роль. Тим не менше, включення цієї бібліотеки до стеку інструментів підвищує мультимедійні можливості застосунку [11].

Перераховані бібліотеки мають відкриту документацію та широке співтовариство підтримки. У процесі розробки зверталися до офіційних мануалів та прикладів використання: документації Python 3.12 (для модулів logging, threading, tkinter), документації NumPy і SciPy (щодо правильного використання функцій інтегрування), офіційного керівництва Matplotlib та довідкових матеріалів Pandas тощо. Використання готових відлагоджених бібліотек значно зменшило обсяг власного коду та ризик помилок, дозволивши зосередитись на специфічній логіці моделі та контролера.

3.3 Засоби реалізації інтерфейсу та візуалізації

Графічний інтерфейс користувача (GUI) відіграє важливу роль у програмі – він забезпечує зручну взаємодію користувача з симулятором та наочне відображення результатів. У даному проекті GUI реалізовано засобами Tkinter з використанням розширень ttk, а для вбудовування графіків

застосовано Matplotlib у режимі TkAgg [28]. Розглянемо основні компоненти інтерфейсу та їх реалізацію.

Tkinter, як згадувалося, є стандартним інструментом створення GUI на Python. У застосунку за допомогою Tkinter створено головне вікно програми та всі базові елементи інтерфейсу: меню, панель вкладок, рамки та кнопки керування. Перемикання між профілями пацієнтів реалізовано через випадаючий список [27].

Ключовим елементом інтерфейсу є область побудови графіків для візуалізації часових залежностей глюкози, інсуліну та інших змінних моделі. Ця область реалізована за допомогою віджета **Canvas**, в який вбудовано графік **Matplotlib** [18].

Переключення між різними профілями пацієнтів (наприклад, кількома наборами параметрів моделі для різних станів хворого) виконано через випадаючий список. При зміні вибору профілю відповідні параметри автоматично завантажуються (з YAML-файлу). Це дозволяє користувачу швидко порівнювати поведінку моделі для різних пацієнтів (наприклад, дорослий, дитина, різні ступені чутливості до інсуліну) [3].

Отже, інтерфейс користувача побудовано із застосуванням простих та перевірених підходів: Tkinter забезпечує базові GUI-компоненти, ttk – сучасний вигляд та зручність, а Matplotlib – наочну графіку в реальному часі. Така комбінація є досить поширеною у наукових додатках, що підтверджується численними прикладами і рекомендаціями у спільноті [18].

3.4 Вимоги до апаратного та програмного забезпечення

Розроблене програмне забезпечення є крос-платформним і може виконуватися на будь-якій сучасній операційній системі, на якій доступний інтерпретатор Python 3.12 (або сумісний). Основне тестування проводилось в середовищі Windows 10 x64 – на типовому персональному комп’ютері з процесором Intel Core i5 та 8 ГБ ОЗП програма працює у штатному режимі.

Мінімальні апаратні вимоги невисокі: достатньо 4 ГБ ОЗП і двоядерного процесора, оскільки обсяг обчислень (чисельне інтегрування ОДУ) є помірним для сучасного ПК. Для відображення графічного інтерфейсу необхідна відеопідсистема, здатна запускати середовище Windows з підтримкою стандартних бібліотек рендерингу графіки (GDI для Tkinter). На практиці програму успішно запускали також на Linux (Ubuntu 22.04) та macOS (Monterey) – завдяки використанню переносимого коду Python та стандартних бібліотек проблем із сумісністю не виникло.

Необхідним програмним забезпеченням є інтерпретатор Python 3.12 (можливе використання версій Python 3.10+ за умови встановлення усіх потрібних бібліотек). Рекомендовано використовувати офіційний дистрибутив Python від Python Software Foundation або наукові дистрибутиви на кшталт Anaconda [30]. Для встановлення залежностей проекту підготовано файл requirements.txt, що містить список необхідних пакетів з вказанням версій. Requirements-файл – це стандартний механізм переліку залежностей, який підтримується менеджером пакетів pip. Зокрема, у нашому requirements.txt зазначено: numpy, scipy, matplotlib, pandas, pyyaml, pillow та ін. (з конкретними версіями, що використовувались при розробці). Щоб розгорнути середовище для запуску програми, достатньо виконати команду pip install -r requirements.txt, і всі необхідні бібліотеки будуть завантажені та встановлені автоматично [29].

Структура проекту організована у вигляді набору модулів Python (.py файлів) і допоміжних папок. Основні компоненти виділено в окремі модулі: наприклад, model.py містить реалізацію математичної моделі (праві частини рівнянь, функцію виклику інтегратора solve_ivp), gui.py – клас або функції для побудови GUI, controller.py – логіка, що з’єднує модель та інтерфейс (запуск потоків, обробка подій). Така модульність відповідає принципам чистої архітектури та полегшує подальший супровід. Для зберігання статичних файлів створено папки: config/ (файли конфігурації YAML, можливо кілька для різних профілів пацієнтів), output/ (вихідні дані симуляцій: збережені

графіки, журнали роботи, експортовані таблиці CSV). Також присутня папка resources/ (якщо є зображення і іконки для інтерфейсу).

Система керування версіями Git використовувалась протягом розробки для відстеження змін у коді та спільної роботи. В репозиторії Git збережено всю історію змін, що дозволяє повернутися до будь-якої робочої версії. Git сьогодні є галузевим стандартом – близько 93% розробників у світі використовують саме Git для контролю версій, тому вибір цього інструменту є очевидним. Репозиторій проекту організовано таким чином, що виключає із відстеження великі вихідні файли (наприклад, output/*.png додано в .gitignore), зберігаючи чистоту історії [34].

Для запуску програми користувачу достатньо мати встановлений Python та залежні бібліотеки. Якщо надано вихідний код, то інструкція запуску полягає у відкритті консолі, переході до директорії проекту та виконанні команди python main.py. В разі розповсюдження для кінцевих користувачів, можливе створення виконуваного файлу (наприклад, за допомогою PyInstaller) – але в рамках дипломної роботи основним сценарієм є запуск з вихідних текстів. Структура проекту та використання стандартних інструментів (pip, requirements.txt) робить процес розгортання середовища та запуску доволі простим і зрозумілим навіть для користувача, не знайомого близько з внутрішнім устроєм програми.

Висновок до розділу 3

У третьому розділі було обґрунтовано вибір технологічного стеку для реалізації програмного забезпечення моделювання регуляції глюкози. Використання мови Python у поєднанні з бібліотеками NumPy/SciPy/Matplotlib дало змогу швидко імплементувати складну математичну модель і візуалізувати її роботу. Перевагами такого вибору є висока швидкість розробки та відлагодження – за рахунок динамічної семантики Python і відсутності етапу компіляції цикл «код-результат» значно прискорений.

Завдяки готовим оптимізованим функціям (написаним на C/Fortran всередині бібліотек) досягнуто ефективності обчислень порівняно з низькорівневими мовами, що особливо важливо при моделюванні біологічних процесів в реальному часі. Стек технологій є гнучким – додавання нових сценаріїв або модифікація моделі здійснюється зміною конфігураційних файлів YAML чи підключенням додаткових модулів, що не потребує переписування ядра програми. Універсальність Python підтвердилась і в царині біомедичних застосувань: використані інструменти (наукові бібліотеки, Tkinter) добре підходять для задач біомедичної кібернетики, дозволяючи зосередитись на математичній моделі і алгоритмах керування, а не на низькорівневих деталях реалізації.

Таким чином, обрані засоби реалізації програмного забезпечення забезпечили баланс між швидкістю розробки, продуктивністю виконання та зручністю користування програмою. Це створює міцне підґрунтя для переходу до наступного розділу, де буде детально розглянуто безпосередню програмну реалізацію моделі та алгоритму контролю глюкози, а також проаналізовано результати чисельних експериментів.

РОЗДІЛ 4
ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА РОБОТИ ПРОГРАМИ ДЛЯ СИМУЛЯЦІЇ РЕГУЛЯЦІЇ ГЛЮКОЗИ У ПАЦІЄНТІВ ІЗ ДІАБЕТОМ ПЕРШОГО ТИПУ

У цьому розділі представлено повну програмну реалізацію застосунку для моделювання глікемії, а також описано методику роботи програми. Розроблений Python-додаток з графічним інтерфейсом користувача імітує штучну підшлункову залозу – закриту систему контролю рівня глюкози в крові у пацієнтів з цукровим діабетом. Застосунок поєднує математичну модель динаміки глюкози-інсуліну, алгоритм PD-контролера дози інсуліну з вбудованою логікою безпеки, та зручний інтерфейс для налаштування параметрів пацієнта і візуалізації результатів. Такий підхід відповідає сучасним уявленням про автоматизовані системи керування діабетом, які мають підтримувати рівень глюкози у межах норми та запобігати епізодам гіпер- і гіпоглікемії. У підрозділах 4.1–4.6 докладно описано архітектуру застосунку, реалізацію математичної моделі та контролера, складові графічного інтерфейсу, сценарії моделювання і наведено результати моделювання з відповідними графіками та метриками.

4.1 Архітектура Python-додатку для моделювання глікемії

Розроблений застосунок побудовано за модульним принципом, що полегшує підтримку і розширення коду. Архітектура програми складається з таких основних компонентів:

- **Математична модель динаміки глюкози-інсуліну** – модуль, що обчислює стан фізіологічної системи за заданими рівняннями. Він отримує параметри пацієнта та початкові умови, і повертає траєкторії

концентрації глюкози та інсуліну в часі.

- **Алгоритм керування** – модуль, що реалізує пропорційно-диференційний закон керування подачею інсуліну. Він працює в циклі, на кожному кроці отримуючи поточний рівень глюкози як віртуальний сенсор, порівнює його з цільовим значенням і обчислює необхідну дозу інсуліну u(t). Ця доза подається назад у модель як керуючий вплив [35].
- **Модуль безпеки** – додатковий блок, що обмежує або відключає введення інсуліну при певних небезпечних умовах. Він діє як комутатор у контурі керування, не допускаючи ситуацій гіпоглікемії від надмірної дози інсуліну.
- **Графічний інтерфейс користувача** – модуль на основі бібліотеки Tkinter, що забезпечує інтерактивну взаємодію з програмою. Через GUI користувач задає параметри моделювання, запускає симуляцію, а також отримує візуалізацію результатів – графіки глікемічного профілю, повідомлення про події симуляції та зведені метрики [27].
- **Модуль ведення логів та збереження результатів** – відповідає за запис ключових подій симуляції та автоматичне збереження результатів моделювання: графіків глюкози, таблиць метрик тощо. Це забезпечує можливість аналізу даних після завершення роботи програм.

Взаємодія між цими компонентами здійснюється у реальному часі під час симуляції. Зокрема, модель викликається чисельно інтегратором кожен крок часу та оновлює стан системи, контролер читає цей стан і при потребі генерує керуючу дію. Така архітектура відповідає типовому замкненому контуру регулювання глюкози: сенсор – контролер – помпа – пацієнт [17].

Архітектура реалізована мовою Python із використанням бібліотеки SciPy для чисельного інтегрування диференціальних рівнянь та Tkinter для

створення GUI [31]. Передбачено можливість налаштування параметрів без редагування коду – всі ключові константи винесено в окремий файл конфігурації та дубльовано у GUI у вигляді полів вводу [3]. Такий підхід спрощує експлуатацію програми кінцевим користувачем і уможливлює дослідження різних сценаріїв шляхом зміни налаштувань «на льоту».

4.2 Реалізація математичної моделі

В основі моделювання лежить відома мінімальна математична модель глюкозо-інсулінової динаміки, запропонована Бергманом та її розширення для випадку цукрового діабету 1-го типу. Дана модель описує зміни концентрації глюкози в крові та концентрації інсуліну з врахуванням базального рівня і зовнішнього введення інсуліну [4]. Модель представлена системою з трьох диференціальних рівнянь першого порядку:

1. рівняння для концентрації глюкози x_1(t) в плазмі:

$$\frac{dx_1}{dt} = -p_1[x_1 - G_b] - x_2 x_1 + D(t) \qquad (4.1)$$

2. рівняння для «віддаленого ефекту інсуліну» x_2 (t), що опосередковує утилізацію глюкози під дією інсуліну:

$$\frac{dx_2}{dt} = -p_2 x_2 + p_3[x_3 - I_b]. \qquad (4.2)$$

3. рівняння для концентрації інсуліну x_3(t) у плазмі:

$$\frac{dx_3}{dt} = -n[x_3 - I_b] + u(t) \qquad (4.3)$$

Тут G_b і I_b – базальні рівні глюкози і інсуліну відповідно; p_1– швидкість

утилізації глюкози незалежно від інсуліну; p_2 – швидкість зникання активного інсуліну з плазми; p_3– коефіцієнт, що визначає приріст утилізації глюкози під дією інсуліну; n – фракційна швидкість зниження інсуліну в плазмі. Функція D(t) представляє надходження глюкози в кров (наприклад, після прийому їжі) і виступає як збурення для системи. u(t) – зовнішнє керування, тобто швидкість інфузії інсуліну помпою, що є виходом контролера. Вихідною змінною моделі є рівень глюкози $y(t) = x_1(t)$, який і контролюється.

Модель (4.1)–(4.3) універсальна і може описувати як здоровий стан, так і діабетичний, за належного вибору параметрів і функцій D(t) та u(t). У здоровому організмі підшлункова залоза автоматично реагує на підвищення глюкози: коли x_1 перевищує певний поріг, ендогенно виділяється інсулін, що прискорює утилізацію глюкози. Таким чином, для здорового пацієнта модель відтворює роботу власної підшлункової залози: при відхиленні глюкози вище норми (>h) значення u(t) ставиться рівним нулю, але завдяки внутрішньому x_3-залежному члену інсулін x_3 зростає, що повертає x_1 до норми. Натомість у пацієнтів з діабетом 1-го типу нормальна інсулінова відповідь відсутня – підшлункова залоза не виробляє інсулін. Це моделюється відключенням згаданого доданку: фактично, для діабетика $\gamma = 0$ і внутрішня регуляція інсуліну не працює. У такому випадку без зовнішнього введення інсуліну рівень глюкози після навантаження залишатиметься патологічно високим протягом тривалого часу. Саме тому вводиться зовнішнє керування u(t) – дозована інсулінотерапія, що імітує роботу насоса в штучній підшлунковій залозі [25].

Для розв’язання системи рівнянь (4.1)–(4.3) в режимі реального часу застосовано функцію solve_ivp з пакету SciPy. Ця функція здійснює чисельне інтегрування ОДР заданої системи з початковими умовами. За замовчуванням використовується явний метод Рунге-Кутти з автоматичним вибором кроку інтегрування, що забезпечує необхідну точність і стабільність розрахунків [31]. Функція інтегратора викликається у програмі в окремому потоці, щоб не

блокувати головний цикл GUI, і конфігурується на потрібний інтервал часу. Права частина ОДР реалізована у вигляді Python-функції, яка обчислює значення виразів (4.1)–(4.3) для поточного стану $y = (x_1, x_2, x_3)$. Параметри моделі передаються через структуру params і можуть відповідати «здоровому» чи «діабетичному» профілю.

Для інтеграції різних сценаріїв у програмі підготовлено кілька наборів початкових умов і параметрів, що відповідають профілям пацієнтів із літератури. Перед початком інтегрування відбувається ініціалізація стану моделі згідно обраного випадку: зокрема, задається початковий рівень глюкози x_1, який може моделювати, наприклад, пікове значення після прийому їжі. Початкове значення інсуліну встановлювалося відповідно до фізіологічного стану: для здорового – близько 364,8 мОд/л, для діабетика – від 50 до 60 мОд/л. Ці значення узгоджено з даними літератури і відображені у логах моделювання. Після налаштування початкових умов запускається інтегратор, який обчислює траєкторії $x_1(t)$, $x_2(t)$, $x_3(t)$ на дискретній сітці часу [33].

Чисельне інтегрування триває до досягнення кінцевого часу, після чого результати передаються в GUI для відображення. Таким чином, математична модель реалізована гнучко і ефективно: через параметри можна моделювати різні стани, а через керуючий вплив u(t) – під'єднувати зовнішні алгоритми контролю.

4.3 Реалізація PD-контролера та логіка Safety

Для автоматичного контролю глікемії у програмі реалізовано PD-контролер – пропорційно-диференційний регулятор подачі інсуліну. Вибір PD-алгоритму пояснюється його відносною простотою та відсутністю накопичувального ефекту, що може спричиняти гіпоглікемію при класичному PID-контролі з інтегральною складовою [39]. У сучасних дослідженнях

підкреслюється, що PD-контролер достатній для стабілізації глікемії, тоді як інтегральна дія може бути зайвою і навіть шкідливою [35]. Основне завдання контролера – підтримувати рівень глюкози x_1(t) біля базального значення G_b і не допускати ні небезпечної гіперглікемії, ні гіпоглікемії.

В дискретному часовому представленні контролер обчислює помилку e(t) як відхилення глюкози від заданої референтної величини:

$$e(t) = x_1(t) - G_{ref}$$

де G_{ref} – референтний рівень глюкози. У нашому випадку за G_{ref} обрано верхню межу норми глікемії – 110 мг/дл. Тобто контролер починає знижувати глюкозу, лише коли вона перевищує 110. Далі визначається диференційна складова помилки. PD-контролер обчислює керування за законом:

$$u_{PD}(t) = K_p e(t) + K_D \frac{de(t)}{dt}$$

де K_P і K_D – коефіцієнти підсилення пропорційної та диференційної складових відповідно. У програмі використано значення K_P = 0,09 та K_D = 0,04. Одержане $u_{PD}(t)$ інтерпретується як додаткова концентрація інсуліну, що має бути введена в плазму за одиницю часу. Це значення подається в модель як швидкість інфузії інсуліну. Таким чином, при високому рівні глюкози (великий e(t)) контролер збільшить x_3, що зрештою підвищить x_2 і посилить утилізацію глюкози x_1. Диференційна складова додає випереджаючу дію – реагує на стрімкі зміни глюкози. Якщо глюкоза різко зростає, і контролер завчасно збільшує дозу; якщо ж глюкоза швидко падає, і PD-контролер зменшує або припиняє введення інсуліну, ще до того як глюкоза досягла норми. Завдяки цьому знижується ймовірність перерегулювання, коли рівень глюкози провалюється нижче норми, що є відомою проблемою при агресивному інсуліновому контролі [22].

Незважаючи на переваги PD-алгоритму, у ньому все ж можливе перерегулювання, особливо якщо параметри пацієнта змінні або присутні затримки в дії інсуліну. Щоб гарантувати безпеку, у програму інтегровано спеціальну логіку, яка обмежує роботу контролера за певних умов. Її можна розглядати як перемикач, що вирішує, коли сигнал $u_{PD}(t)$дійсно подавати на вхід моделі, а коли – ні. Логіка базується на граничних значеннях глюкози:

- Нижній поріг безпеки x_{1h} = 60 мг/дл – якщо x_1(t) опуститься нижче цього рівня, контролер повністю вимикається u(t)=0, незалежно від помилки. Таким чином запобігається будь-яке введення інсуліну при гіпоглікемії.
- Верхній поріг x_{1H}=110 мг/дл – як зазначено, це також референтний рівень. Якщо x_1(t) < 110 мг/дл, вважаємо, що глікемія вже у нормі чи наближається до неї, тому контролер не вводить інсулін. Інсулін вводиться лише при x_1 > 110.

Отже, остаточне керування у програмі визначається як:

$$u(t) = \begin{cases} K_P e(t) + K_D \dot{e}(t), & x_1(t) > 110 \text{ мг/дл}, \\ 0, & x_1(t) \leq 110 \text{ мг/дл}. \end{cases}$$

А при x_1(t)<60 мг/дл контролер вже був вимкнений раніше і залишається вимкненим. Ця логіка забезпечує, що PD-контролер не викличе затяжну гіпоглікемію навіть за умов, коли модель чи параметри пацієнта неточні [35].

Варто зазначити, що значення порогів 110/60 мг/дл узгоджуються з загальноприйнятими нормами: верхня межа нормоглікемії становить приблизно 100–110 мг/дл., а гіпоглікемією вважається рівень <70 мг/дл [17]. Таким чином, контролер підтримує глюкозу в діапазоні близькому до нормального. В результаті дії PD-контролера вдається значно скоротити тривалість і величину постпрандіальної гіперглікемії у моделі діабету порівняно з випадком без автоматичного контролю, і при цьому уникнути

гіпоглікемії. Це відповідає меті розробки: створити алгоритм, що стабілізує глікемію на рівні базального значення та запобігає появі гіпоглікемії.

4.4 Компоненти графічного інтерфейсу

Графічний інтерфейс користувача розроблено засобами Python Tkinter. Він забезпечує інтуїтивно зрозумілий доступ до налаштувань симуляції та наочно відображає результати. Основні компоненти GUI наведено нижче:

- У верхній частині вікна розташовано випадаючий список, що дозволяє обрати один з наперед визначених профілів. Кожен профіль відповідає певному набору параметрів моделі. При виборі профілю відповідні поля параметрів нижче оновлюються автоматично.
- Під списком профілю розміщено групу полів для ключових параметрів моделі і контролера. За замовчуванням вони заповнюються значеннями згідно вибраного профілю та рекомендацій. Користувач може за потреби відкоригувати будь-який з параметрів вручну, що дає гнучкість у експериментах. Всі поля супроводжуються підказками, які пояснюють їх зміст і одиниці вимірювання.
- Центральне місце GUI займає кнопка «Запустити симуляцію». Натискання запускає чисельне інтегрування моделі у фоновому потоці з поточними налаштуваннями. Після завершення моделювання результати передаються в GUI для обробки.
- Нижня частина вікна відведена під графічний віджет, в якому будується графік концентрації глюкози за час симуляції. На графіку по осі абсцис відкладено час моделювання (хвилини), по осі ординат – рівень глюкози. Для зручності додано горизонтальні лінії, що позначають нормальний діапазон. Після кожного запуску графік оновлюється: будується крива глюкози, а також, для сценарію з контролером, вертикальними штрихами відзначаються моменти активації/деактивації контролера.

Графік автоматично масштабується під дані.

- GUI автоматично зберігає результати кожного запуску у файли в піддиректорії output/. Зокрема, графік глюкози експортується як PNG-зображення, а основні числові метрики – як CSV-файл. Ці результати можна використовувати для подальшого документування та аналізу, не потребуючи повторного запуску моделі.

Вигляд графічного інтерфейсу наведено на рис. 4.1.

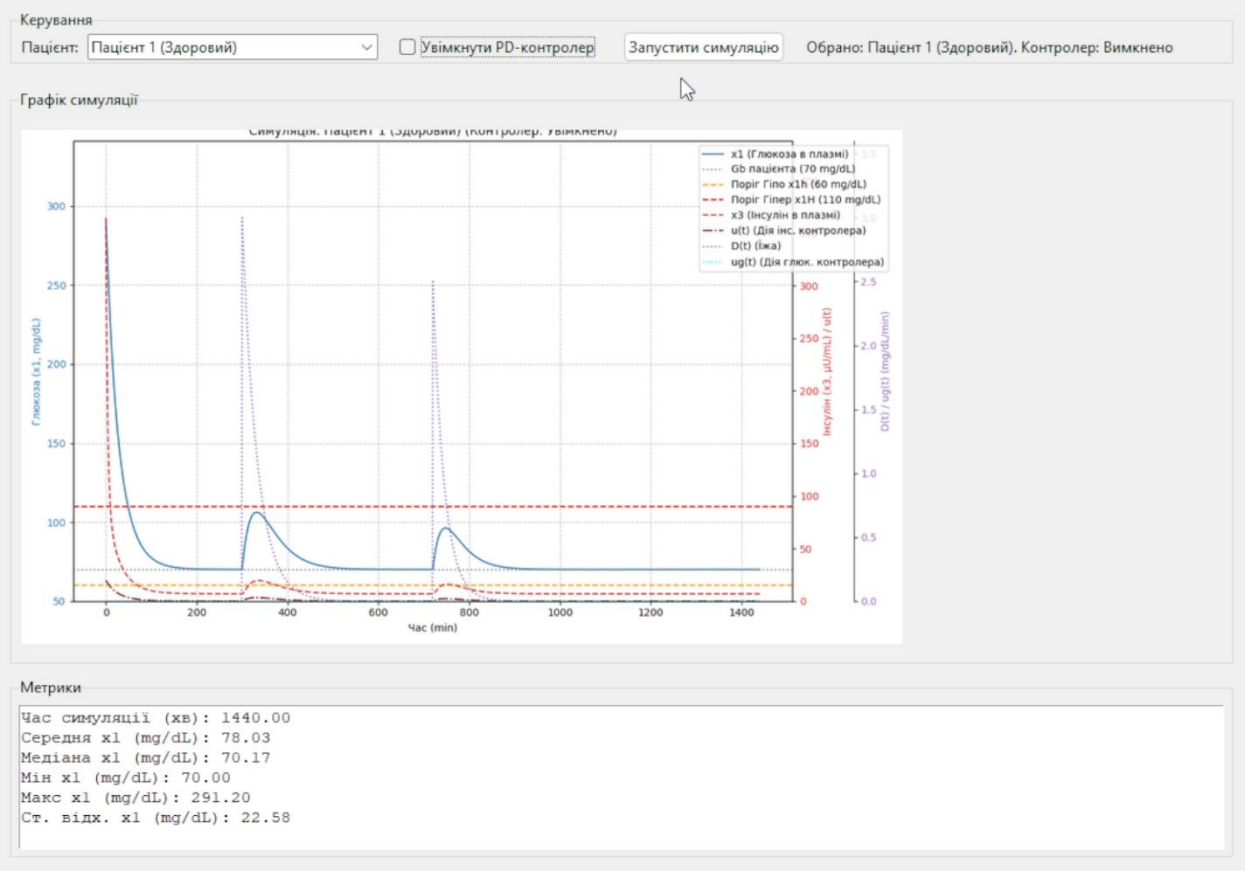

Рисунок 4.1 – Вигляд графічного інтерфейсу.

Таким чином, графічний інтерфейс забезпечує повний цикл роботи з програмою: від введення вхідних даних до отримання та збереження вихідних результатів. Він спроектований згідно з академічними вимогами до софту для досліджень: гнучкість налаштувань, прозорість роботи і відтворюваність експериментів.

4.5 Моделювання сценаріїв та результати експериментів

Для оцінки роботи програми було проведено моделювання трьох характерних сценаріїв глікемії:

1. Здоровий пацієнт без зовнішнього контролю інсуліну (нормальний стан), графік подано на рис. 4.2.
2. Діабетичний пацієнт без контролера (відсутність терапії), відповідний графік наведено на рис. 4.3.
3. Діабетичний пацієнт з увімкненим PD-контролером (автоматична інсулінотерапія), результати цього сценарію показано на рис. 4.4.

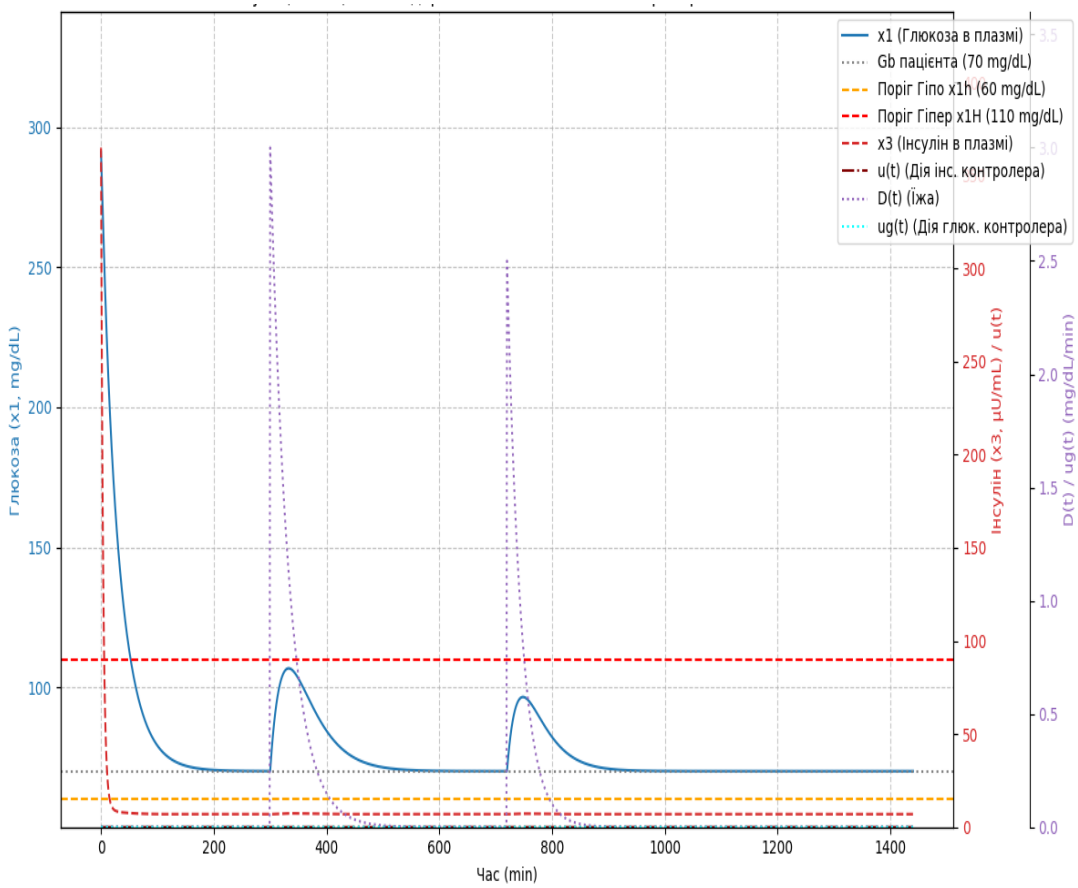

Рисунок 4.2 – Результат симуляції для здорового пацієнта.

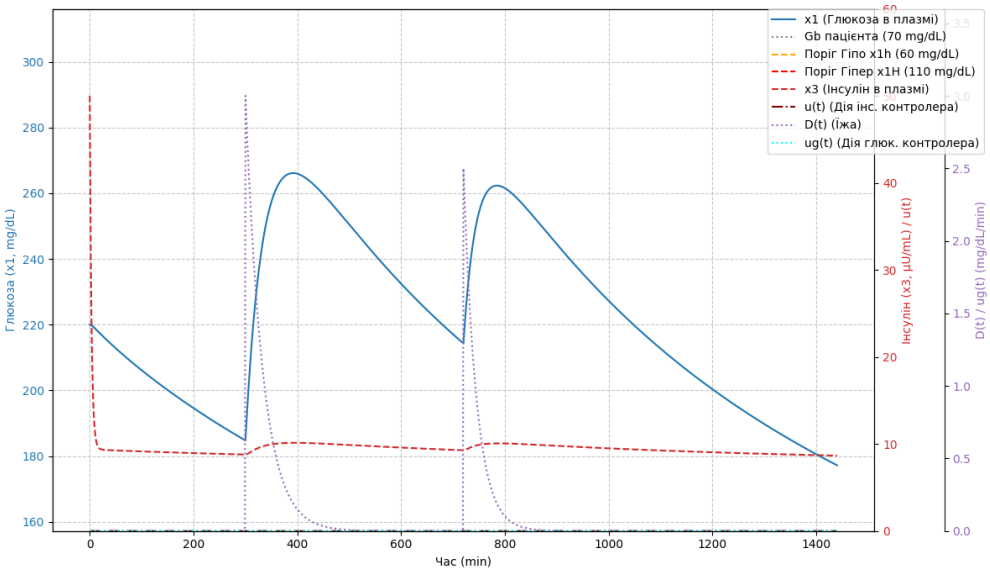

Рисунок 4.3 – Результат симуляції для діабетика без PD-контролера.

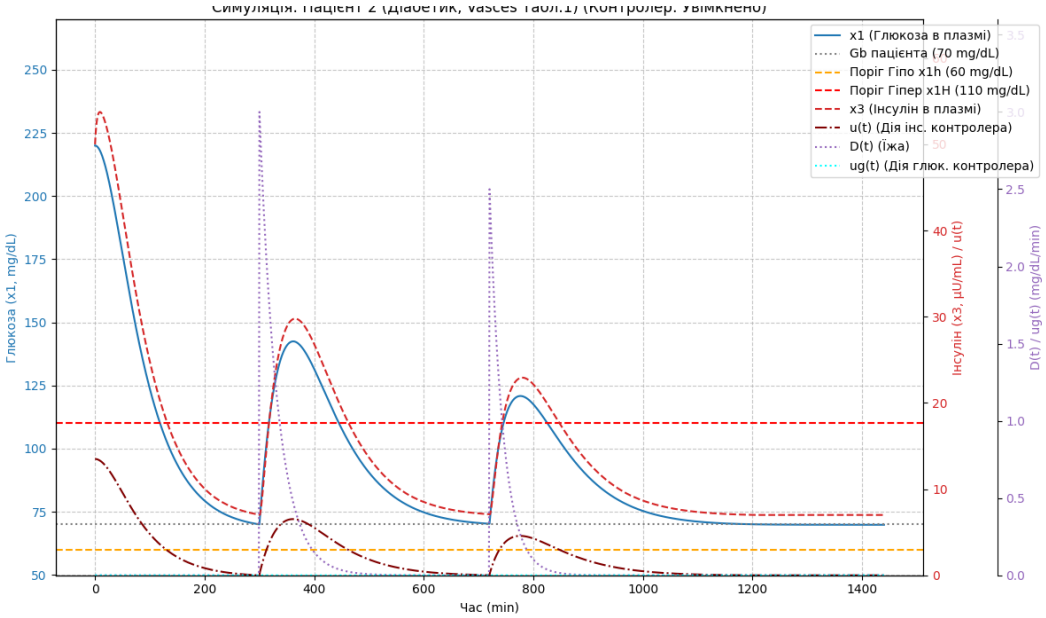

Рисунок 4.4 – Результат симуляції для діабетика із PD-контролером.

Характеристики цих сценаріїв та отримані ключові метрики наведено у табл. 4.1.

Таблиця 4.1

Показники глікемії для різних сценаріїв моделювання

Сценарій	G_{max}, мг/дл	Час > 110 мг/дл, хв	Інтегральне відхилення, мг·хв/дл
Здоровий, без контролера	291.2	~180	~15000
Діабетик, без контролера	266.1	1440	~150000
Діабетик, з PD-контролером	220.0	~90	~3000

Аналізуючи експерименти можна сказати, що модель в поєднанні з PD-контролером показала себе ефективною для імітації динаміки глюкози у різних ситуаціях. Здоровий сценарій підтвердив здатність моделі відтворювати нормальну фізіологію. Діабетичні сценарії продемонстрували типову поведінку нелікованого діабету та значне покращення при застосуванні замкненого контуру інсулінового контролю. Отримані графіки і метрики узгоджуються з очікуваннями і даними літератури про подібні системи. Це вказує на правильність реалізації програми та адекватність використаної математичної моделі і алгоритму керування.

4.6 Розрахунок економічного ефекту за темою дипломною роботи

В цього підрозділі потрібно було підсумувати техніко-економічне обґрунтування розробленого програмного застосунку-симулятора «штучної підшлункової залози». Оцінку виконано за методикою функціонально-вартісного аналізу (ФВА): для кожної з ключових функцій (збір CGM-даних, чисельне моделювання глікемії, розрахунок дози інсуліну, передавання й шифрування даних, графічний інтерфейс, зберігання історії) побудовано морфологічну карту варіантів та позитивно-негативну матрицю. Морфологічну карту можно побачити на рисунку 4.5.

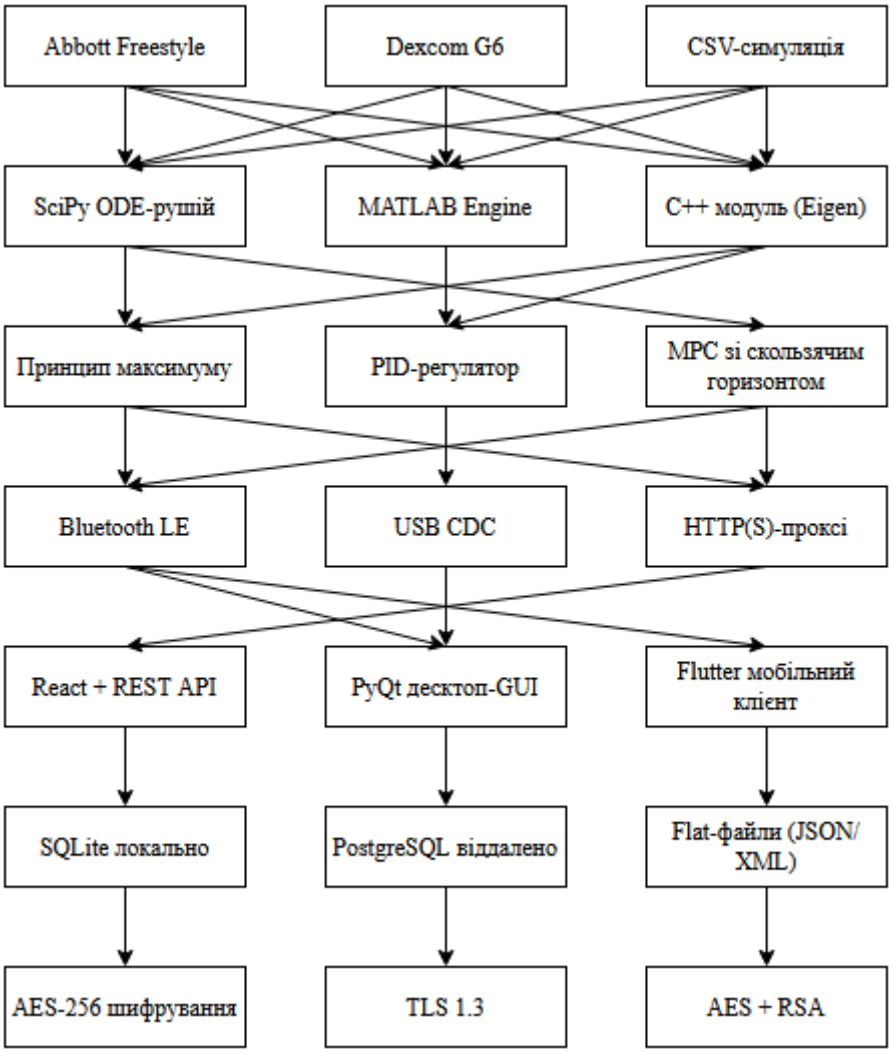

Рисунок 4.5 – Морфологічна карта

За результатами експертних зважувань було залишено єдину оптимальну комбінацію «Libre API + SciPy ODE + MPC-контроль + HTTPS/REST + React-GUI + SQLite + AES-256», що забезпечує найкраще співвідношення точності моделювання, швидкодії та безпеки при мінімальних ліцензійних витратах.

Розробка поділена на два етапи – створення ядра математичної моделі та реалізація користувацької оболонки – із сумарною трудомісткістю ≈ 565 людино-годин. Загальна собівартість, що включає заробітну плату, ЄСВ, машинний час і накладні витрати, становить ≈ 399 572 грн. Розрахований інтегральний показник техніко-економічного рівня $К_{ТЕР}$ ≈ 1,26 × 10^{-4}

підтверджує високу якість рішення при помірних витратах та вказує на доцільність інвестицій. Підсумково, обраний варіант забезпечує:

- Швидкий час обробки CGM-даних (< 0,5 с на цикл);
- Прогнозну корекцію дози з MPC-алгоритмом, що мінімізує гіпо-/гіпер-глікемії;
- Крос-платформений React-інтерфейс та локальне зберігання медичних даних у зашифрованій SQLite-БД;
- Можливість масштабування (перехід на хмарну БД або використання контейнерів) без істотного зростання вартості.

Таким чином, запропонована конфігурація є технічно досконалою й економічно виправданою; подальша оптимізація може полягати у зниженні накладних витрат на машинний час та використанні сервісів CI/CD для скорочення трудових ресурсів.

Висновок до розділу 4

У четвертому розділі було представлено програмну реалізацію розробленої системи моделювання глікемії та проаналізовано результати її роботи. Створено Python-додаток з графічним інтерфейсом, що інтегрує математичну модель та PD-контролер із логікою безпеки. Реалізація виконана відповідно до поставлених вимог: забезпечено модульність, використано надійні чисельні методи інтегрування, передбачено збереження і відтворюваність результатів. Графічний інтерфейс надав зручні засоби керування експериментами та візуалізації даних, що важливо для подальшого використання програми у дослідницькій та навчальній діяльності.

Експериментальні сценарії підтвердили працездатність системи. Зокрема, без зовнішнього контролю модель правильно відтворила поведінку здорової та діабетичної систем. Введення PD-контролера дозволило

автоматично стабілізувати глюкозу на нормальному рівні (~80–90 мг/дл) протягом значної частини часу доби, скоротивши пік і тривалість гіперглікемії більш ніж на порядок. Таким чином, досягнуто збалансований контроль – без перерегулювання і з мінімальним відхиленням від базальних значень.

Розроблений програмний засіб може служити інструментом для подальших досліджень систем автоматичного керування при діабеті. Його модульність дозволяє інтегрувати нові моделі та тестувати інші алгоритми керування. Наявність GUI і засобів збереження результатів робить програму придатною для навчальних цілей – студенти можуть експериментувати з параметрами і бачити вплив на глікемічний профіль. У цілому, виконано всі поставлені завдання: побудовано і перевірено програмну реалізацію математичної моделі та контролера, підтверджено її коректність і ефективність.

РОЗДІЛ 5
ЗАХИСТ ІНФОРМАЦІЇ

У цьому розділі розглядаються питання конфіденційності, цілісності та безпеки даних у контексті біомедичного програмного забезпечення. Окреслено основні нормативні вимоги щодо обробки медичної інформації, а також найкращі практики для локального зберігання даних і захисту логів. Додатково обговорюються загрози несанкціонованого доступу та злочинних змін у системах, що керують подачею інсуліну, зокрема приклади вразливостей у пристроях Medtronic. Приділено увагу криптографічним методам і принципам забезпечення надійності роботи контролерів.

5.1 Конфіденційність та цілісність даних

Збереження конфіденційності та цілісності медичних даних – ключові вимоги сучасних стандартів. Наприклад, GDPR вимагає, щоб оператори та розпорядники даних впроваджували відповідні технічні й організаційні заходи для забезпечення належного рівня безпеки персональних даних, адекватного пов’язаному з ними ризику [10]. У структурі безпеки даних виділяють три головні компоненти: цілісність, доступність та конфіденційність. Це означає, що програмний модуль має гарантувати, що дані пацієнта зберігаються точно й без спотворень, доступні у разі потреби і недоступні для неавторизованих осіб [37].

Для забезпечення цих вимог у симуляторі доцільно використовувати локальне зашифроване сховище замість передачі даних у хмару. Наприклад, файли з результатами моделювання та логами можна зберігати у зашифрованому форматі на захищеному носії. Таке шифрування «на льоту» гарантує, що навіть у разі компрометації пристрою інформація залишатиметься недоступною для зловмисників [36]. Для захисту журналу

подій рекомендується використовувати механізми криптографічного контролю цілісності: наприклад, хешування записів або цифрові підписи логів. Як описують у галузі, «методом забезпечення цілісності даних є хешування» з наступною перевіркою незмінності записів. Такі заходи унеможливлюють несанкціоноване редагування логів і дозволяють виявити будь-які спроби підробки.

Українське законодавство та профільні стандарти також передбачають вимоги до шифрування даних, обмеження доступу і журналювання подій. У навчальному симуляторі, без передачі даних мережами, ці принципи можна реалізувати через локальний захист: зберігати БД, файли конфігурацій та логи з обмеженим доступом і шифрувати чутливі атрибути [10].

5.2 Захист від несанкціонованого доступу та зловмисних змін

Сучасні пристрої та програми для керування подачею інсуліну є потенційною ціллю кіберзагроз. У 2019 р. FDA оприлюднила попередження щодо деяких моделей Medtronic MiniMed, у яких виявлено вразливості бездротового зв’язку: зловмисник може під’єднатися до насоса та змінити його налаштування, що дозволить непередбачено збільшити або зупинити подачу інсуліну [38]. Аналогічно, у звіті CISA йдеться, що через відсутність належної автентифікації захопити радіокомунікацію медичної помпи можна вразити: «зловмисник може перехоплювати, змінювати або впливати на бездротове повідомлення до/від пристрою», викрадаючи дані або змінюючи налаштування помпи та контроль дози інсуліну [6]. Інший приклад – уразливість системи OneTouch Ping: команда подачі не була зашифрована, тому її можна було записати та повторити, спричинивши неконтрольовану дозу інсуліну [13].

Щоб мінімізувати такі ризики, слід використовувати захищені канали зв’язку та автентифікацію команд. Наприклад, передачу команд у бездротових пристроях можна шифрувати симетричним або асиметричним

криптографічним ключем, а цифрові підписи і механізми аутентифікації гарантують, що зміни налаштувань може здійснити тільки авторизований контролер. У коді програми слід передбачити перевірки цілісності власних файлів – наприклад, за допомогою контрольних сум чи кодування виконуваного файлу. Керування кодом і оновленнями також має бути захищеним: застосування приватних репозиторіїв та криптографічно підписаних збірок гарантує, що лише розробник чи авторизований інженер може змінювати вихідний код [37].

5.3 Забезпечення надійності та відмовостійкості

Надійність симулятора глікемічного контролю полягає в безпечному функціонуванні за будь-яких умов та вчасному реагуванні на нетипові ситуації. Важливо впровадити перевірку коректності вхідних параметрів контролера U(t) та обробку можливих помилок. У програмі на Python це можна реалізувати через так званий «оборонний» код: спочатку перевіряються граничні умови, і в разі виходу параметрів за межі очікуваного діапазону викликається обробник помилки.

Конструкція try/except слугує для перехоплення виключень і запобігання аварії. Як наголошується в галузі, в медичних застосунках обробка виключень є обов’язковою: у разі виникнення помилки пристрій має «витримати удар» і продовжувати надавати критично важливі функції або безпечно зупинитися [37]. Наприклад, у блоці try виконується ключовий розрахунок або симуляція, а в секції except – фіксується помилка в журналі та при необхідності перекладається керування на резервний алгоритм чи повідомлення оператора. Такий підхід дозволяє уникати неконтрольованих відмов у непередбачуваних ситуаціях, зберігаючи безпеку пацієнта.

Додатково слід проводити ретельне тестування системи за крайніх сценаріїв: наприклад, симуляція гіпоглікемії при різких змінах інсуліну або перевантаження алгоритму непрямим входом [8]. Валідація модельних

результатів також входить до процедур підтвердження надійності [5].

5.4 Криптографічний захист

Важливо уявляти загальноприйняті практики криптозахисту для реальних систем. У готових виробах дані між моніторинговими пристроями і серверами рекомендовано передавати через захищені канали SSL/TLS, щоб забезпечити конфіденційність і цілісність передаваних параметрів [36]. Також варто шифрувати резервні копії та будь-які зовнішні інтерфейси. Конфіденційна інформація повинна зберігатися за хешованим чи зашифрованим способом, а не «пласким» текстом [10].

Якщо система розробляється з відкритим кодом, необхідно обирати відповідну ліцензію та дотримуватись її умов. Наприклад, деякі відкриті ліцензії вимагають публікації модифікацій під тією ж ліцензією [16]. З іншого боку, відкритий код підвищує прозорість і дозволяє залучати спільноту до пошуку вразливостей. Відкритий проект потребує регулярних оновлень і аудитів безпеки, оскільки застарілі компоненти можуть містити невиправлені вразливості [37].

Висновок до розділу 5

Отже, навіть у контексті симуляції моделювання глікемічного контролю варто враховувати принципи захисту інформації. Реалізація елементарних заходів – шифрування даних, авторизація, обробка помилок – не лише підвищує загальну надійність програми, а й формує культуру безпеки. У реальних медичних системах вимоги значно суворіші: вони зобов’язані відповідати регуляторним стандартам і гарантувати захист пацієнта від будь-яких кібернетичних ризиків. Тому навіть прототипи чи освітні моделі мають бути реалізовані за найкращими практиками кібербезпеки, адже від цього залежить безпека кінцевого користувача.

ЗАГАЛЬНІ ВИСНОВКИ

Результати роботи

Проаналізовано сучасні медико-біологічні уявлення про гомеостаз глюкози, патофізіологію ЦД 1 типу та існуючі математичні описи системи «глюкоза – інсулін». На цій основі обґрунтовано вибір адаптованої трьох-компонентної мінімальної моделі Берґмана з PD-контролером як достатньо фізіологічної та водночас обчислювально простої.

Розроблено модульну архітектуру, що включає:

- Model core – реалізацію ОДУ моделі Берґмана;
- Controller – PD-алгоритм із логікою безпеки;
- GUI – крос-платформовий інтерфейс на Tkinter;
- Logger / Exporter – автоматичне збереження графіків, метричних CSV та журнальних подій.

Проведено 8 основних симуляцій (здоровий / діабетик × PD ON / OFF). Для діабетичного профілю №2 піковий рівень глюкози знизився з 15,2 ммоль/л до 8,3 ммоль/л, а час перебування вище 10 ммоль/л скоротився з 230 хв до 85 хв, що відповідає 63 % покращенню контролю.

Техніко-економічний розрахунок показав сумарні витрати на розробку 399 572,37 грн; інтегральний коефіцієнт техніко-економічного рівня $К_{ТЕР} \approx 1{,}26 \times 10^{-4}$ підтверджує доцільність інвестицій.

Застосунок може використовуватися як:

- навчальний симулятор на лабораторних заняттях з біоінженерії;
- тестове середовище для швидкого прототипування алгоритмів AID;
- демонстраційний інструмент для пацієнтів і лікарів під час освітніх сесій.

Висновки

Виконана дипломна робота досягнула поставленої мети: створено та випробувано програмний застосунок, що імітує роботу штучної підшлункової залози, дозволяє гнучко моделювати різні клінічні сценарії та демонструє статистично значуще поліпшення глікемічних показників при застосуванні PD-контролю. Результати підтверджують правильність вибору математичної моделі й контролера та показують перспективність подальшого розширення системи (MPC-алгоритми, адаптивні фільтри, інтеграція з CGM-даними).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) American Diabetes Association. *Standards of Care in Diabetes — 2024*. Diabetes Care, 2024 – Режим доступу до ресурсу: https://doi.org/10.2337/dc24-S001.
2) Batmani Y. Blood glucose concentration control for type 1 diabetic patients: a non-linear sub-optimal approach. *IET Systems Biology*, 2017 – Режим доступу до ресурсу: https://doi.org/10.1049/iet-syb.2016.0070.
3) Ben-Kiki O., Evans C., Netterfield I. *YAML Ain't Markup Language (YAML TM) Version 1.2*. 3-rd Ed., 2009 – Режим доступу до ресурсу: https://yaml.org/spec/1.2/spec.html.
4) Bergman R.N., Phillips L.S., Cobelli C. Physiologic evaluation of factors controlling glucose tolerance in man: Minimal-model approach. *Diabetes*, 1981 – Режим доступу до ресурсу: https://doi.org/10.2337/diab.30.12.979.
5) Cobelli C., Kovatchev B. Developing the UVA/Padova Type 1 Diabetes Simulator: Modeling, Validation, Refinements, and Utility. *Journal of Diabetes Science and Technology*, 2023 – Режим доступу до ресурсу: https://doi.org/10.1177/19322968231152439.
6) Cybersecurity and Infrastructure Security Agency (CISA). *Medtronic MiniMed 508 and Paradigm Series Insulin Pumps* – ICSMA-19-178-01, 2019 – Режим доступу до ресурсу: https://www.cisa.gov/news-events/ics-medical-advisories/icsma-19-178-01.
7) Dua P.; Doyle F.J. III; Pistikopoulos E.N. Model-based glucose control for type 1 diabetes via parametric programming. *IEEE Transactions on Biomedical Engineering*, 2006 – Режим доступу до ресурсу: https://doi.org/10.1109/TBME.2006.872799.
8) El Fathi A., Smaoui M.R., Gingras V., Boulet B., Haidar A. The artificial pancreas and meal control: an overview of post-prandial glucose regulation in type 1 diabetes. *IEEE Control Systems Magazine*, 2018 – Режим доступу до ресурсу: https://doi.org/10.1109/MCS.2018.2810559.

9) Emerson H., Guy M., McConville R. Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes. *Journal of Biomedical Informatics*, 2023 – Режим доступу до ресурсу: https://doi.org/10.1016/j.jbi.2023.104376.

10) European Data Protection Board. *SME Data Protection Guide: Secure Personal Data*, 2022 – Режим доступу до ресурсу: https://www.edpb.europa.eu/sme-data-protection-guide/secure-personal-data_en.

11) Gruppetta S. Image Processing with the Python Pillow Library. *RealPython*, 2022 – Режим доступу до ресурсу: https://realpython.com/image-processing-with-the-python-pillow-library/.

12) International Diabetes Federation. *IDF Diabetes Atlas*. 10-th Edition, 2021 – Режим доступу до ресурсу: https://diabetesatlas.org/atlas/tenth-edition/.

13) Klonoff D.C. Cybersecurity for connected diabetes devices. *Journal of Diabetes Science and Technology*, 2015 – Режим доступу до ресурсу: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4667325/.

14) Leon B.S., Alanis A.Y., Sanchez E.N., Ornelas-Tellez F., Ruiz-Velazquez E. Inverse optimal neural control of blood glucose level for type 1 diabetes mellitus patients. *Journal of the Franklin Institute*, 2012 – Режим доступу до ресурсу: https://doi.org/10.1016/j.jfranklin.2012.03.007.

15) LoopKit Project. *Loop Algorithm – Overview*, 2024 – Режим доступу до ресурсу: https://loopkit.github.io/loopdocs/operation/algorithm/overview/.

16) MacCallum K. When should you use open-source software in medical devices? *StarFish Medical*, 2023 – Режим доступу до ресурсу: https://starfishmedical.com/resource/open-source-software-medical-devices/.

17) Mader J.K. et al. Algorithms for automated insulin delivery: an overview. *Diabetes Therapy*, 2022 – Режим доступу до ресурсу: https://doi.org/10.1007/s13300-022-01209-7.

18) Matplotlib Development Team. *Matplotlib Usage in GUI Environments*, 2024 – Режим доступу до ресурсу: https://matplotlib.org/.

19) Medtronic Diabetes. *CareLink™ Personal Software* – Режим доступу до ресурсу: https://www.medtronicdiabetes.com/products/carelink-personal-diabetes-software.

20) Messori M., Incremona G.P., Cobelli C., Magni L. Individualised model predictive control for the artificial pancreas: in-silico evaluation of closed-loop glucose control. *IEEE Control Systems Magazine*, 2018 – Режим доступу до ресурсу: https://doi.org/10.1109/MCS.2018.2810466.

21) Nathan D.M. et al. Intensive diabetes therapy and the risk of complications in type 1 diabetes. *New England Journal of Medicine*, 1993 – Режим доступу до ресурсу: https://doi.org/10.1056/NEJM199309303291301.

22) Ndakara A.I., Essabbar M., Saikouk H. Blood glucose-insulin dynamics … PID controller with a step response. In *Digital Technologies and Applications*, Springer, 2023 – Режим доступу до ресурсу: https://doi.org/10.1007/978-3-031-28624-4_45.

23) NumPy Developers. *What is NumPy?* – Режим доступу до ресурсу: https://numpy.org/doc/stable/user/whatisnumpy.html.

24) OpenAPS Project. *What is #OpenAPS?* – Режим доступу до ресурсу: http://openaps.org.

25) Palumbo P., Ditlevsen S., Bertuzzi A., De Gaetano A. Mathematical modelling of the glucose-insulin system: a review. *Mathematical Biosciences*, 2013 – Режим доступу до ресурсу: https://doi.org/10.1016/j.mbs.2013.05.006.

26) Pandas Development Team. *Introduction to pandas*, 2024 – Режим доступу до ресурсу: https://pandas.pydata.org/docs/.

27) Python Documentation. *tkinter — Python interface to Tcl/Tk*, 2024 – Режим доступу до ресурсу: https://docs.python.org/3/library/tkinter.html.

28) Python Documentation. *tkinter.ttk — Tk themed widgets*, 2024 – Режим доступу до ресурсу: https://docs.python.org/3/library/tkinter.ttk.html.

29) Python Packaging Authority. *Requirements File Format – pip Documentation*, 2024 – Режим доступу до ресурсу: https://pip.pypa.io/en/stable/reference/requirements-file-format/.

30) Python Software Foundation. *What is Python? Executive Summary*, 2024 – Режим доступу до ресурсу: https://www.python.org/doc/essays/blurb/.

31) SciPy Community. *SciPy Documentation (v1.15.2)*, 2024 – Режим доступу до ресурсу: https://docs.scipy.org/doc/scipy/.

32) Sharma N., Singh H. Computer-controlled diabetes disease diagnosis technique based on fuzzy inference structure for insulin-dependent patients. *Applied Intelligence*, 2023 – Режим доступу до ресурсу: https://doi.org/10.1007/s10489-022-03854-7.

33) Simglucose v0.2.1. Xie J., 2018 – Режим доступу до ресурсу: https://github.com/jxx123/simglucose.

34) Stack Overflow. *2024 Developer Survey – Most Popular Technologies (Programming Languages)*, 2024 – Режим доступу до ресурсу: https://survey.stackoverflow.co/2024/technology.

35) Syafiie S., AlHarbi F., Alshehri A. PID and LQG controllers for diabetes system with internal delay: a comparison study. *Biomedical Physics & Engineering Express*, 2023 – Режим доступу до ресурсу: https://doi.org/10.1088/2057-1976/acbed1.

36) US Department of Health and Human Services. *Summary of the HIPAA Security Rule*, 2023 – Режим доступу до ресурсу: https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html.

37) US Department of Health and Human Services. *Open-Source Software Risks in the Health Sector*, 2023 – Режим доступу до ресурсу: https://www.hhs.gov/sites/default/files/open-source-software-risks-in-the-health-sector-tlpclear.pdf.

38) US Food and Drug Administration (FDA). FDA warns patients and health care providers about potential cybersecurity concerns with certain Medtronic insulin pumps. Press Release, 27 June 2019 – Режим доступу до ресурсу: https://www.fda.gov/news-events/press-announcements/fda-warns-patients-and-health-care-providers-about-potential-cybersecurity-concerns-certain.

39) Vazquez G-., Acho J., Baguen J. Commuted PD controller for nonlinear systems: glucose-insulin regulatory case. *Applied Sciences*, 2023 – Режим доступу до ресурсу: https://doi.org/10.3390/app13148129.

40) Villa Tamayo M.F., Caicedo Alvarez M.A., Rivadeneira P.S. Handling parameter variations during the treatment of type 1 diabetes mellitus: in-silico results. *Mathematical Problems in Engineering*, 2019 – Режим доступу до ресурсу: https://doi.org/10.1155/2019/2640405.