

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова  
праця на правах рукопису

**ЦАПІВ ЯРОСЛАВ АНДРІЙОВИЧ**

УДК 004.41:004.946:519.6:615.851.82

**ДИСЕРТАЦІЯ**

**МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ  
РЕАБІЛІТАЦІЙНОЇ АРТ-ТЕРАПІЇ ВЕРХНІХ КІНЦІВОК ПАЦІЄНТІВ  
НА ОСНОВІ AR-ТЕХНОЛОГІЙ**

Спеціальність 121 – Інженерія програмного забезпечення  
Галузь знань 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії.

Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне джерело

  
\_\_\_\_\_ Я.А. Цапів

Науковий керівник: Пукас Андрій Васильович, доктор технічних наук, професор

Тернопіль – 2026

## АНОТАЦІЯ

*Цанів Я.А.* Математичне та програмне забезпечення реабілітаційної арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 – Інженерія програмного забезпечення – Західноукраїнський національний університет, Тернопіль, 2026.

Підготовка здійснювалась на кафедрі комп'ютерних наук Західноукраїнського національного університету Міністерства освіти і науки України.

У дисертаційній роботі розв'язано науково-технічне завдання розробки математичного та програмного забезпечення системи реабілітаційної арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій, що розширило функції програмного комплексу для автоматичного об'єктивного контролю динаміки відновлення рухливості суглобів, прогнозування результатів реабілітації та адаптивної корекції терапевтичного протоколу. Метою дисертаційного дослідження є підвищення ефективності реабілітації верхніх кінцівок пацієнтів у спосіб розроблення математичного та програмного забезпечення системи реабілітаційної арт-терапії на основі AR-технологій та автоматичної гоніометрії.

У вступі наведено актуальність теми дисертаційного дослідження, мету та основні завдання, а також наукову новизну та практичну значущість.

У першому розділі проведено детальний аналіз існуючих методів і програмного забезпечення для вимірювання кутів суглобів, технологій VR/AR у реабілітації, методів прогнозування динаміки відновлення та систем рекомендацій у фізичній терапії. Розглянуто контактні методи гоніометрії (механічні гоніометри, цифрові інклінометри, IMU-системи), маркерні системи захоплення руху (Vicon, OptiTrack) та безмаркерні системи оцінки пози на базі глибокого навчання (OpenPose, BlazePose, MediaPipe Pose Landmarker, MoveNet, ViTPose, YOLOv8-pose). Встановлено, що контактні методи мають недостатню точність (5–10°),

маркерні системи є непрактичними через високу вартість, а існуючі безмаркерні системи потребують адаптивної фільтрації шуму та багатокамерної тріангуляції. Проаналізовано систематичні огляди та мета-аналізи ефективності VR/AR у реабілітації верхніх кінцівок. Показано, що класичні методи прогнозування (регресійний аналіз, ARIMA, нейронні мережі) мають суттєві обмеження при застосуванні до коротких клінічних вибірок. Порівняльний аналіз існуючих VR/AR-реабілітаційних платформ виявив критичний розрив: жодна система не інтегрує одночасно безмаркерну гоніометрію, інтервальне прогнозування, AR-арт-терапію та рекомендаційну систему у замкнений адаптивний цикл.

У заключній частині першого розділу, на підставі проведеного аналізу та виявлених недоліків існуючих рішень, здійснено постановку науково-технічного завдання. Сформульовано мету та основні задачі дисертаційного дослідження. Обґрунтовано доцільність застосування інтервального підходу з ідентифікацією на основі алгоритму бджолоїної колонії для прогнозування динаміки відновлення на коротких клінічних вибірках.

У другому розділі наведено результати розробки математичних моделей та методів безмаркерної гоніометрії і динаміки відновлення кутів рухливості суглобів. Розроблено метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків, який базується на нейромережевому детекторі YOLOv8-pose, адаптивній фільтрації координат OneEuro-фільтром та алгоритмі геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху. Запропонований метод забезпечує середню абсолютну похибку менше  $4^\circ$  для всіх типів рухів плечового суглоба, що знаходиться в межах мінімальної клінічно значущої різниці.

У цьому ж розділі на основі методів структурної та параметричної ідентифікації інтервальних дискретних динамічних моделей із застосуванням алгоритму бджолоїної колонії побудовано моделі динаміки відновлення кутів рухливості суглобів. Структурна ідентифікація забезпечує вибір оптимального набору базисних функцій, а параметрична – знаходження інтервальних оцінок

коефіцієнтів моделі. Побудовані моделі забезпечують стовідсоткове покриття експериментальних спостережень та клінічно прийнятну точність прогнозування.

У завершальній частині розділу описано адаптивний цикл прогнозування та корекції терапевтичного протоколу, що реалізує замкнений контур керування реабілітаційним процесом: виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів.

У третьому розділі розкрито принципи проектування програмного забезпечення комплексної AR-системи реабілітації. Спроектовано мікросервісну архітектуру комплексної програмної системи AR-реабілітації верхніх кінцівок, побудовану на принципі оркестрації мікросервісів: мікросервісу аналізу трьох відеопотоків з пріоритизацією на Python, мікросервісу інтервального прогнозування та мікросервісу рекомендацій на основі генеративного штучного інтелекту, оркестрованих центральним сервером на NestJS із базою даних PostgreSQL, а також AR-застосунку арт-терапії для гарнітури Magic Leap 2 на Unity та вебзастосунку фізіотерапевта на React. Описано загальну архітектуру системи з чотирма незалежними каналами даних між мікросервісами, програмну архітектуру мікросервісу автоматичної гоніометрії з конвеєрною обробкою відеопотоків, архітектуру AR-застосунку арт-терапії та архітектуру серверної частини з рекомендаційним мікросервісом. Основні компоненти функціонують у локальній мережі реабілітаційного кабінету; зовнішнє інтернет-з'єднання використовується лише рекомендаційним модулем для генерації терапевтичних зображень.

У четвертому розділі наведено результати реалізації та клінічної апробації розробленої системи. Описано реалізацію мікросервісу автоматичної гоніометрії, AR-застосунку арт-терапії для гарнітури Magic Leap 2, серверної частини з рекомендаційним мікросервісом та вебзастосунку фізіотерапевта. На клінічних даних п'яти реабілітаційних пацієнтів КНП «Тернопільська обласна клінічна лікарня» ТОР побудовано інтервальні математичні моделі динаміки відновлення кутів плечового суглоба для трьох ключових рухів: згинання, розгинання та відведення. На прогнозній групі з двох пацієнтів проведено апробацію адаптивного циклу прогнозування та корекції терапевтичного протоколу.

Результати апробації підтвердили ефективність розроблених рішень: інтервальна модель виявила неоптимальну динаміку відновлення вже після 3 сеансів, рекомендаційна система запропонувала зміну терапевтичного зображення, після корекції середній темп приросту кута згинання зріс у 1,9–3,9 раза, а прогнозована кількість сеансів до нормативного діапазону скоротилася на 12%. Отримані результати підтверджують функціональну завершеність комплексної системи AR-реабілітації та практичну реалізацію замкненого контуру керування реабілітаційним процесом.

У висновках сформульовано основні результати дисертаційної роботи.

Результати дисертаційної роботи впроваджено у процес проведення реабілітаційних сеансів КНП «Тернопільська обласна клінічна лікарня» ТОР, у науково-дослідну роботу кафедри фізичної терапії Тернопільського національного медичного університету імені І.Я. Горбачевського, у виробничий процес ТОВ «ЕЛЕКС» та у освітній процес кафедри комп'ютерних наук Західноукраїнського національного університету, що підтверджується відповідними актами впровадження.

**Ключові слова:** програмне забезпечення, архітектура програмних систем, комп'ютерний зір, нейронна мережа (YOLO), штучний інтелект (ШІ, машинне навчання), глибоке навчання, доповнена реальність (AR, віртуальне середовище, віртуальна лабораторія), гарнітура (AR-гарнітура, нейрогарнітура), інтервальні моделі (інтервальні дані, аналіз інтервальних даних, модель), ідентифікація (структурна ідентифікація, параметрична ідентифікація), поведінкові моделі штучної бджолиної колонії (штучна бджолина колонія), прогнозування, прийняття рішень, реабілітація (фізична реабілітація, фізична терапія), діагностика (функціональна діагностика).

## **ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ**

*Наукові праці, в яких опубліковано основні наукові результати дисертації*

1. Dyvak M, Tsapiv Y, Pukas A, Petrovskyi Y, Melnyk A, Dyvak A, Banasik A, Czupryna-Nowak A, Pikiewicz P, Popuk Y, Dzyha Y. AR Technology for Restoring Upper

Limb Joint Mobility in Patients. Applied Sciences, 2026, Vol. 17. 5878. ISSN: 2076-3417. (3,5 д.а. / 2 д.а.; особистий внесок: розробка комплексної архітектури AR-системи реабілітації, інтеграція компонентів у замкнутий адаптивний цикл прогнозування та корекції терапевтичного протоколу, проведення клінічного дослідження).

DOI: [10.3390/app16125878](https://doi.org/10.3390/app16125878) (Scopus, Q2)

2. Цапів Я.А., Тихий Р.Р. Інноваційна вебсистема гоніометричного аналізу на базі нейронних мереж. Наукові праці ДонНТУ. Серія “Інформатика, кібернетика та обчислювальна техніка”, 2025, № 1 (40), с. 80–88. (кат. Б) ISSN 1996-1588. (0,9 д.а. / 0,7 д.а.; особистий внесок: розробка програмної архітектури вебсистеми гоніометричного аналізу, реалізація модулів нейромережевої детекції анатомічних точок та обчислення кутів суглобів).

DOI: [10.31474/1996-1588-2025-1-40-80-88](https://doi.org/10.31474/1996-1588-2025-1-40-80-88)

3. Пукас А.В., Цапів Я.А. Автоматизована безмаркерна система вимірювання діапазонів рухів суглобів на основі трикамерного відеоаналізу. Наукові праці ДонНТУ. Серія “Інформатика, кібернетика та обчислювальна техніка”, 2026, № 1 (42), с. 54–64. (кат. Б) ISSN 1996-1588. (1,1 д.а. / 0,9 д.а.; особистий внесок: розробка методу безмаркерного трикамерного вимірювання кутів суглобів, реалізація конвеєрної архітектури підсистеми відеоаналізу).

DOI: [10.31474/1996-1588-2026-1-42-54-64](https://doi.org/10.31474/1996-1588-2026-1-42-54-64)

4. Цапів Я.А., Пукас А.В., Біловус Д. Рекомендаційна система підтримки фізичної реабілітації на основі корекції терапевтичних зображень. Вимірювальна та обчислювальна техніка в технологічних процесах, 2026. № 2, с.387-396. (кат. Б). ISSN 2219-9365. (1 д.а. / 0,9 д.а.; особистий внесок: розробка рекомендаційної системи підтримки фізичної реабілітації, реалізація багатоступеневого AI-контурної генерації арт-терапевтичного контенту).

DOI: [10.31891/2219-9365-2026-86-45](https://doi.org/10.31891/2219-9365-2026-86-45)

*Наукові праці, які засвідчують апробацію матеріалів дисертації*

5. Tsapiv Y., Dyvak M., Popovych D., Havrylenko A., Tykhyi R. Kasianchuk V. Automated Goniometric Analysis System Based on Neural Networks, 2025 15th International Conference on Advanced Computer Information Technologies (ACIT), Sibenik, Croatia, 2025, pp. 762-766. ISSN 2770-5218. (0,5 д.а. / 0,3 д.а.; особистий внесок: розробка методу автоматичного гоніометричного аналізу на базі нейронних мереж, реалізація алгоритмів адаптивної фільтрації координат та геометричного злиття вимірювань від трьох камер).

DOI: [10.1109/ACIT65614.2025.11185897](https://doi.org/10.1109/ACIT65614.2025.11185897) (Scopus).

6. Tykhyi R., Tsapiv Ya., Dyvak M., Popovych D., Havrylenko A., Husieva T. Software system for supporting art therapy processes using augmented reality technologies. Proceedings - International Conference on Advanced Computer Information Technologies, ACIT, 2024, p. 660–664. ISSN 2770-5218. (0,5 д.а. / 0,3 д.а.; особистий внесок: проектування та реалізація AR-застосунку арт-терапії для гарнітури Magic Leap 2, розробка модулів просторового позиціонування, тактильної компенсації та рендерингу текстур малювання).

DOI: [10.1109/ACIT62333.2024.10712584](https://doi.org/10.1109/ACIT62333.2024.10712584) (Scopus).

7. Головка Р.А., Шостопаль В.А., Цапів Я.А. Математичне та програмне забезпечення для організації процесу тестування web-орієнтованих систем. Комп'ютерні інформаційні технології: Матеріали школи-семінару молодих вчених і студентів СІТ'2023. – Тернопіль: ЗУНУ, с.90. (0,1 д.а. / 0,03 д.а.; особистий внесок: розробка архітектури програмного забезпечення для тестування).

8. Цапів Я.А. Архітектура програмної системи арт-терапії та реабілітації з AR. Комп'ютерні інформаційні технології: Матеріали зимової школи-семінару молодих вчених і студентів СІТ'2025. – Тернопіль: ЗУНУ, с. 57–58.

9. Цапів Я.А. Адаптивний цикл реабілітації верхніх кінцівок на основі AR-технологій. Матеріали весняної школи-семінару молодих вчених і студентів СІТ'2025. – Тернопіль: ЗУНУ, с. 78–79.

## ANNOTATION

*Tsapiv Y. A.* Mathematical and software support for rehabilitation art therapy of patients' upper limbs based on AR technologies. – Scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy in the specialty 121 – Software Engineering – West Ukrainian National University, Ternopil, 2026.

The research was carried out at the Department of Computer Science of the West Ukrainian National University, Ministry of Education and Science of Ukraine.

The dissertation addresses the scientific and technical task of developing mathematical and software support for a rehabilitation art therapy system for patients' upper limbs based on AR technologies, which extends the functions of the software complex for automatic objective control of joint mobility recovery dynamics, prediction of rehabilitation outcomes, and adaptive correction of the therapeutic protocol. The aim of the dissertation research is to improve the effectiveness of upper limb rehabilitation through the development of mathematical and software support for a rehabilitation art therapy system based on AR technologies and automatic goniometry.

The introduction presents the relevance of the dissertation topic, its aim and main objectives, as well as the scientific novelty and practical significance.

The first chapter provides a detailed analysis of existing methods and software for joint angle measurement, VR/AR technologies in rehabilitation, methods for predicting recovery dynamics, and recommendation systems in physical therapy. Contact goniometry methods (mechanical goniometers, digital inclinometers, IMU systems), marker-based motion capture systems (Vicon, OptiTrack), and markerless pose estimation systems based on deep learning (OpenPose, BlazePose, MediaPipe Pose Landmarker, MoveNet, ViTPose, YOLOv8-pose) were considered. It was established that contact methods have insufficient accuracy (5–10°), marker-based systems are impractical due to high cost, and existing markerless systems require adaptive noise filtering and multi-camera triangulation. Systematic reviews and meta-analyses of

VR/AR effectiveness in upper limb rehabilitation were analyzed. It was shown that classical forecasting methods (regression analysis, ARIMA, neural networks) have significant limitations when applied to short clinical samples. A comparative analysis of existing VR/AR rehabilitation platforms revealed a critical gap: no system simultaneously integrates markerless goniometry, interval prediction, AR art therapy, and a recommendation system into a closed adaptive cycle.

In the concluding part of the first chapter, based on the analysis and identified shortcomings of existing solutions, the scientific and technical problem was formulated. The aim and main objectives of the dissertation research were defined. The feasibility of applying the interval approach with identification based on the artificial bee colony algorithm for predicting recovery dynamics on short clinical samples was justified.

The second chapter presents the results of developing mathematical models and methods for markerless goniometry and joint mobility recovery dynamics. A method of markerless goniometry based on the analysis of three video streams was developed, which is based on the YOLOv8-pose neural network detector, adaptive coordinate filtering with the OneEuro filter, and an algorithm for geometric fusion of measurements from three orthogonally positioned cameras with prioritization based on movement planes. The proposed method provides a mean absolute error of less than  $4^\circ$  for all types of shoulder joint movements, which is within the minimal clinically important difference.

In the same chapter, based on the methods of structural and parametric identification of interval discrete dynamic models using the artificial bee colony algorithm, models of joint mobility recovery dynamics were constructed. Structural identification ensures the selection of an optimal set of basis functions, while parametric identification provides interval estimates of model coefficients. The constructed models ensure 100% coverage of experimental observations and clinically acceptable prediction accuracy.

In the concluding part of the chapter, an adaptive cycle of prediction and correction of the therapeutic protocol was described, implementing a closed-loop control of the rehabilitation process: execution – measurement – prediction – recommendation – image correction – movement adaptation.

The third chapter reveals the principles of software design for the comprehensive AR rehabilitation system. A microservice architecture of the comprehensive AR rehabilitation software system for upper limbs was designed, based on the principle of microservice orchestration: a Python-based microservice for three-video-stream analysis with prioritization, a microservice for interval forecasting, and a microservice for recommendations based on generative artificial intelligence, orchestrated by a central server on NestJS with a PostgreSQL database, as well as an AR art therapy application for the Magic Leap 2 headset on Unity and a physiotherapist web application on React. The overall system architecture with four independent data channels between microservices, the software architecture of the automatic goniometry microservice with pipeline video stream processing, the architecture of the AR art therapy application, and the architecture of the server component with the recommendation microservice were described. The main components operate within the local network of the rehabilitation room; external internet connectivity is used only by the recommendation module for generating therapeutic images.

The fourth chapter presents the results of implementation and clinical validation of the developed system. The implementation of the automatic goniometry subsystem, AR art therapy application for the Magic Leap 2 headset, server component with recommendation subsystem, and physiotherapist web application was described. Using clinical data from five rehabilitation patients at the Ternopil Regional Clinical Hospital, interval mathematical models of shoulder joint mobility recovery dynamics were constructed for three key movements: flexion, extension, and abduction. The adaptive cycle of prediction and correction of the therapeutic protocol was validated on a prognostic group of two patients.

The validation results confirmed the effectiveness of the developed solutions: the interval model detected suboptimal recovery dynamics after just 3 sessions, the recommendation system proposed a change of the therapeutic image, and after correction the average rate of flexion angle increase grew by 1.9–3.9 times, while the predicted number of sessions to reach the normative range decreased by 12%. The obtained results

confirm the functional completeness of the comprehensive AR rehabilitation system and the practical implementation of the closed-loop control of the rehabilitation process.

The conclusions summarize the main results of the dissertation.

The results of the dissertation were implemented in the rehabilitation process at the Ternopil Regional Clinical Hospital, in the research activities of the Department of Physical Therapy of Ternopil National Medical University named after I.Ya. Horbachevsky, in the production process of ELEKS LLC, and in the educational process of the Department of Computer Sciences of West Ukrainian National University, as confirmed by the corresponding implementation certificates.

**Keywords:** software, software systems architecture, computer vision, neural network (YOLO), artificial intelligence (AI, machine learning), deep learning, augmented reality (AR, virtual environment, virtual laboratory), headset (AR headset, neuroheadset), interval models (interval data, interval data analysis, model), identification (structural identification, parametric identification), behavioral models of artificial bee colony (artificial bee colony), predicting, decision-making, rehabilitation (physical rehabilitation, physical therapy), diagnostics (functional diagnostics).

## LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

### *Scientific papers containing the main scientific results of the dissertation*

1. Dyvak M, Tsapiv Y, Pukas A, Petrovskyi Y, Melnyk A, Dyvak A, Banasik A, Czupryna-Nowak A, Pikiewicz P, Popyk Y, Dzyha Y. AR Technology for Restoring Upper Limb Joint Mobility in Patients. Applied Sciences, 2026, Vol. 17. 5878. ISSN 2076-3417. (3.5 p.s. / 2 p.s.; personal contribution: development of a comprehensive AR rehabilitation system architecture, integration of components into a closed adaptive cycle for prediction and correction of the therapeutic protocol, conducting a clinical study).

DOI: 10.3390/app16125878 (Scopus, Q2).

2. Tsapiv Y., Tykhyi R. Innovative web system for goniometric analysis based on neural networks. Scientific Papers of Donetsk National Technical University. Series

“Computer Science, Cybernetics and Computer Engineering”, 2025, No. 1 (40), pp. 80–88. (cat. B). ISSN 1996-1588. (0.9 p.s. / 0.7 p.s.; personal contribution: development of the software architecture for a goniometric analysis web system, implementation of neural network modules for anatomical landmark detection and joint angle computation).

DOI: 10.31474/1996-1588-2025-1-40-80-88.

3. Pukas A., Tsapiv Y. Automated markerless system for measuring joint ranges of motion based on three-camera video analysis. Scientific Papers of Donetsk National Technical University. Series “Computer Science, Cybernetics and Computer Engineering”, 2026, No. 1 (42), pp. 54–64. (cat. B). ISSN 1996-1588. (1.1 p.s. / 0.9 p.s.; personal contribution: development of a markerless three-camera joint angle measurement method, implementation of a pipeline architecture for the video analysis subsystem).

DOI: 10.31474/1996-1588-2026-1-42-54-64.

4. Tsapiv Y., Pukas A., Bilovus D. Recommendation system for physical rehabilitation support based on therapeutic image correction. Measurement and Computing Technology in Technological Processes, 2026, No. 2, pp. 387–396 (cat. B). ISSN 2219-9365. (1 p.s. / 0.9 p.s.; personal contribution: development of a recommendation system for physical rehabilitation support, implementation of a multi-stage AI pipeline for art-therapeutic content generation).

DOI: 10.31891/2219-9365-2026-86-45.

*Scientific papers certifying the approbation of dissertation materials*

5. Tsapiv Y., Dyvak M., Popovych D., Havrylenko A., Tykhyi R., Kasianchuk V. Automated Goniometric Analysis System Based on Neural Networks. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT), Sibenik, Croatia, 2025, pp. 762–766. ISSN 2770-5218. (0.5 p.s. / 0.3 p.s.; personal contribution: development of an automated goniometric analysis method based on neural networks, implementation of adaptive coordinate filtering algorithms and geometric fusion of measurements from three cameras).

DOI: 10.1109/ACIT65614.2025.11185897 (Scopus).

6. Tykhyi R., Tsapiv Ya., Dyvak M., Popovych D., Havrylenko A., Husieva T. Software system for supporting art therapy processes using augmented reality technologies. Proceedings – International Conference on Advanced Computer Information Technologies, ACIT, 2024, pp. 660–664. ISSN 2770-5218. (0.5 p.s. / 0.3 p.s.; personal contribution: design and implementation of an AR art-therapy application for the Magic Leap 2 headset, development of spatial positioning, tactile compensation, and painting texture rendering modules).

DOI: 10.1109/ACIT62333.2024.10712584 (Scopus).

7. Holovko R., Shostopal V., Tsapiv Y. Mathematical and software support for organizing the testing process of web-oriented systems. Computer Information Technologies: Proceedings of the Young Scientists and Students Workshop CIT'2023. – Ternopil: WUNU, p. 90. (0.1 p.s. / 0.03 p.s.; personal contribution: development of the software architecture for web-based system testing).

8. Tsapiv Y. Architecture of a software system for art therapy and rehabilitation with AR. Computer Information Technologies: Proceedings of the Winter Young Scientists and Students Workshop CIT'2025. – Ternopil: WUNU, pp. 57–58.

9. Tsapiv Y. Adaptive cycle of upper limb rehabilitation based on AR technologies. Proceedings of the Spring Young Scientists and Students Workshop CIT'2025. – Ternopil: WUNU, pp. 78–79.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ.....	16
ВСТУП.....	17
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ АВТОМАТИЧНОГО ВИМІРЮВАННЯ КУТІВ СУГЛОБІВ ТА ПРОГНОЗУВАННЯ ДИНАМІКИ РЕАБІЛІТАЦІЇ.....	25
1.1. Аналіз систем вимірювання кутів суглобів та VR/AR-технологій у фізичній реабілітації.....	26
1.2. Аналіз методів вимірювання кутів суглобів та безмаркерних систем оцінки пози.....	32
1.3. Аналіз методів побудови моделей динаміки та прогнозування результатів реабілітації.....	42
1.4. Аналіз систем підтримки клінічних рішень та рекомендаційних систем у фізичній реабілітації.....	53
1.5. Постановка задачі дослідження.....	63
Висновки до розділу 1.....	66
РОЗДІЛ 2 МЕТОДИ БЕЗМАРКЕРНОЇ ГОНІОМЕТРІЇ ТА МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІКИ ВІДНОВЛЕННЯ КУТІВ РУХЛИВОСТІ СУГЛОБІВ.....	68
2.1. Метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків.....	69
2.2. Інтервальні моделі динаміки відновлення кутів рухливості суглобів верхніх кінцівок.....	88
2.3. Адаптивний цикл прогнозування та корекції терапевтичного протоколу...	96
Висновки до розділу 2.....	104
РОЗДІЛ 3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ AR- СИСТЕМИ АРТ-ТЕРАПІЇ.....	106
3.1. Загальна архітектура системи та потоки даних між компонентами.....	107
3.2. Мікросервіс автоматичної гоніометрії.....	112
3.3. Застосунок арт-терапії для гарнітури Magic Leap 2.....	118

3.4. Центральний сервер-оркестратор з мікросервісом рекомендацій та вебзастосунок фізіотерапевта.....	120
Висновки до розділу 3.....	132
<b>РОЗДІЛ 4 РЕАЛІЗАЦІЯ І АПРОБАЦІЯ AR-СИСТЕМИ АРТ-ТЕРАПІЇ ДЛЯ РЕАБІЛІТАЦІЇ ВЕРХНІХ КІНЦІВОК.....</b>	<b>134</b>
4.1. Реалізація мікросервісу автоматичної гоніометрії.....	135
4.2. Реалізація AR-застосунку арт-терапії для гарнітури Magic Leap 2.....	148
4.3. Реалізація центрального сервера-оркестратора з мікросервісом рекомендацій та вебзастосунку фізіотерапевта.....	152
4.4. Інтервальні моделі динаміки відновлення кутів плечового суглоба.....	159
4.5. Апробація адаптивного циклу прогнозування та корекції терапевтичного протоколу.....	169
Висновки до розділу 4.....	179
<b>ВИСНОВКИ.....</b>	<b>181</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>185</b>
<b>ДОДАТОК А РЕАЛІЗАЦІЯ МАТЕМАТИЧНОГО АПАРАТУ АДАПТИВНОЇ ФІЛЬТРАЦІЇ ТА ОБЧИСЛЕННЯ КУТІВ.....</b>	<b>200</b>
<b>ДОДАТОК Б КОНВЕЄР БЕЗМАРКЕРНОГО ЗАХОПЛЕННЯ ТА НЕЙРОМЕРЕЖЕВОГО РЕЗУЛЬТАТУ.....</b>	<b>207</b>
<b>ДОДАТОК В REST API ТА ПОТОКОВА ПЕРЕДАЧА ВІДЕОДАНИХ.....</b>	<b>212</b>
<b>ДОДАТОК Г РЕЗУЛЬТАТИ ІДЕНТИФІКАЦІЇ ІНТЕРВАЛЬНОЇ МОДЕЛІ ДИНАМІКИ ВІДНОВЛЕННЯ КУТА ЗГИНАННЯ ПЛЕЧОВОГО СУГЛОБА....</b>	<b>220</b>
<b>ДОДАТОК Г РЕЗУЛЬТАТИ ІДЕНТИФІКАЦІЇ ІНТЕРВАЛЬНОЇ МОДЕЛІ ДИНАМІКИ ВІДНОВЛЕННЯ КУТА РОЗГИНАННЯ ПЛЕЧОВОГО СУГЛОБА .....</b>	<b>223</b>
<b>ДОДАТОК Д РЕЗУЛЬТАТИ ІДЕНТИФІКАЦІЇ ІНТЕРВАЛЬНОЇ МОДЕЛІ ДИНАМІКИ ВІДНОВЛЕННЯ КУТА ВІДВЕДЕННЯ ПЛЕЧОВОГО СУГЛОБА</b>	<b>226</b>
<b>ДОДАТОК Е СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ .....</b>	<b>229</b>
<b>ДОДАТОК Є АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ.....</b>	<b>231</b>

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

АБК	алгоритм бджолоїної колонії (Artificial Bee Colony)
AAOS	American Academy of Orthopaedic Surgeons – Американська академія ортопедичних хірургів
ARIMA	Autoregressive Integrated Moving Average – авторегресійна інтегрована модель ковзного середнього
CNN	Convolutional Neural Network – згорткова нейронна мережа
COCO	Common Objects in Context – набір даних Microsoft для оцінки пози людини
FPS	Frames Per Second – кадрів на секунду
HPE	Human Pose Estimation – оцінка пози людини
IMU	Inertial Measurement Unit – інерціальний вимірювальний блок
LLM	Large Language Model – велика мовна модель
MAE	Mean Absolute Error – середня абсолютна похибка
ONNX	Open Neural Network Exchange – відкритий формат обміну нейронними мережами
P2P	Peer-to-Peer – однорангове з'єднання
REST	Representational State Transfer – архітектурний стиль взаємодії компонентів
RMS	Root Mean Square – середньоквадратичне значення
ROM	Range of Motion – діапазон рухливості суглоба
SDK	Software Development Kit – набір засобів розробки
WebRTC	Web Real-Time Communication – технологія потокової передачі даних у реальному часі
WebSocket	протокол двостороннього зв'язку між клієнтом і сервером у реальному часі
YOLOv8-pose	You Only Look Once v8 pose – нейромережевий детектор пози людини

## ВСТУП

**Актуальність теми дослідження.** Зростаючі потреби в реабілітаційних послугах є глобальним викликом сучасної охорони здоров'я. За даними Всесвітньої організації охорони здоров'я, понад 1 мільярд людей на планеті потребують реабілітаційних послуг [1], причому нейром'язово-скелетні порушення залишаються провідною причиною інвалідності у всіх регіонах світу [2]. Для України ця проблема набуває особливої гостроти в умовах збройного конфлікту, який спричинив різке збільшення кількості осіб із порушеннями опорно-рухового апарату, зокрема верхніх кінцівок, що потребують тривалої та систематичної фізичної реабілітації.

Об'єктивне вимірювання кутів рухливості суглобів (ROM – Range of Motion) є ключовим показником ефективності реабілітаційних програм [3]. Традиційні методи оцінки ґрунтуються на ручному вимірюванні за допомогою механічних гоніометрів із точністю 5–10° [4], цифрових інклінометрів [5] та IMU-систем (Werium, MPU 6050) [5, 6, 8]. Маркерні системи захоплення руху (Vicon, OptiTrack) вважаються золотим стандартом [9, 10, 11], однак їхня висока вартість робить їх непрактичними для широкого клінічного використання. Системи безмаркерної оцінки пози (HPE) на базі глибокого навчання, зокрема OpenPose (Cao та ін. [12]), BlazePose [13], MediaPipe Pose Landmarker [14], MoveNet [15], ViTPose (Xu та ін. [16]) та YOLOv8-pose (Jocher та ін. [17]), відкрили можливість автоматичного відстежування рухів, проте основними викликами залишаються шум у координатах, проекційна похибка та недостатня деталізація для дрібних суглобів [18]. Для подолання шуму нейромережевої детекції застосовують методи адаптивної фільтрації, зокрема адаптивний фільтр Калмана [19] та OneEuro-фільтр [20], а для усунення проекційної похибки використовують багатокamerну триангуляцію [21, 22], однак ці підходи рідко поєднуються у практичних реабілітаційних системах.

Технології віртуальної (VR) та доповненої реальності (AR) суттєво змінюють підходи до фізичної реабілітації. Систематичні огляди та мета-аналізи Laver та ін.

[23], Bharathi та ін. [24], Maier та ін. [25], Karamians та ін. [26], Peng та ін. [27], Mekbib та ін. [28] підтверджують, що VR/AR-реабілітація статистично значимо покращує функцію верхніх кінцівок порівняно з конвенційною терапією. Особливо перспективним є поєднання VR/AR з арт-терапією (Haeyen та ін. [29], Kaimal та ін. [30], Lo та ін. [31], Kongkasuwan та ін. [32]), яка активує додаткові мотиваційні центри мозку та покращує як фізичні, так і психологічні результати лікування. Проте інтеграція точних автоматичних вимірювань ROM з AR-терапією та математичним прогнозуванням динаміки відновлення залишається практично недослідженою.

Прогнозування динаміки реабілітації є ключовим для переходу від реактивного до проактивного управління терапевтичним процесом. Класичні стохастичні методи (регресія, ARIMA, нейронні мережі [35, 36, 94]) потребують великих обсягів даних, тоді як клінічні вибірки обмежені 10-20 сеансами. Альтернативним підходом є методи аналізу інтервальних даних (Дивак, Мельник, Порплиця [38, 103, 104]) із застосуванням алгоритму бджолої колонії (Karaboga [39]) та його паралельної реалізації (Кіндзерський [40]), які природно працюють із короткими вибірками та обмеженою невизначеністю. Інтеграція таких прогнозів із системами підтримки клінічних рішень (Sutton та ін. [41], Shortliffe та Sepulveda [42]) дозволяє реалізувати замкнутий адаптивний цикл управління реабілітацією.

Порівняльний аналіз існуючих систем - Vidnova [43], VR арт-терапії для інсульту (Tierl та ін. [44]), RAPAEI Smart Glove (Neofect), Tyromotion VirtualRehab, MindMotion GO (MindMaze) - виявив критичний розрив: жодна система не інтегрує одночасно безмаркерну багатоканальну гоніометрію з адаптивною фільтрацією, AR арт-терапію, прогнозування динаміки на основі інтервальних моделей та систему рекомендацій для фізіотерапевта.

Таким чином, розробка математичного та програмного забезпечення AR-системи арт-терапії для реабілітації верхніх кінцівок, яка забезпечить об'єктивне автоматичне вимірювання кутів суглобів, прогнозування динаміки відновлення та адаптивну корекцію терапевтичного протоколу, є актуальним науково-практичним

завданням, що має значний потенціал для клінічного впровадження та трансформації практики фізичної реабілітації.

**Мета і завдання дослідження.** Метою дисертаційного дослідження є підвищення ефективності реабілітації верхніх кінцівок пацієнтів у спосіб розроблення математичного та програмного забезпечення системи реабілітаційної арт-терапії на основі AR-технологій та автоматичної гоніометрії.

Для досягнення зазначеної мети встановлено такі завдання дослідження:

1. провести аналіз існуючих методів і програмного забезпечення для фізичної реабілітації, зокрема на основі VR/AR технологій, вимірювання кутів суглобів, методів прогнозування динаміки відновлення та систем рекомендацій у фізичній терапії;

2. розробити метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією координат ключових анатомічних точок та алгоритмом геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху;

3. побудувати інтервальні математичні моделі динаміки відновлення кутів рухливості плечового суглоба на основі адаптованих методів структурної та параметричної ідентифікації з використанням алгоритму бджолиної колонії;

4. спроектувати комплексну архітектуру медичної реабілітаційної програмної системи на основі оркестрації мікросервісів: аналізу трьох відеопотоків з пріоритизацією, прогнозування, рекомендацій на основі генеративного штучного інтелекту та застосунку арт-терапії для гарнітури Magic Leap 2, а також вебзастосунку фізіотерапевта;

5. інтегрувати розроблені компоненти – автоматичну безмаркерну гоніометрію, інтервальне прогнозування та систему рекомендацій на основі генеративного штучного інтелекту – в єдиний замкнутий адаптивний цикл прогнозування та корекції терапевтичного протоколу;

6. реалізувати програмну систему, провести апробацію на реальних даних реабілітації пацієнтів та клінічне дослідження ефективності розроблених методів.

**Об'єкт дослідження** – процеси реабілітації верхніх кінцівок пацієнтів із застосуванням арт-терапевтичних методик.

**Предмет дослідження** – математичне та програмне забезпечення реабілітаційної арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій, автоматичної гоніометрії та інтервальних математичних моделей динаміки відновлення рухливості суглобів.

**Методи дослідження.** У роботі використано: методи адаптивної фільтрації сигналів (фільтр OneEuro) - для зменшення шуму у координатах ключових анатомічних точок; методи обчислювальної геометрії (триангуляція, геометрична фузія) - для перетворення 2D-координат від трьох відеопотоків у 3D-координати суглобів з пріоритизацією на основі площин руху; методи інтервального аналізу та теорії ідентифікації - для побудови інтервальних дискретних динамічних моделей відновлення рухливості суглобів; метаевристичний алгоритм на основі поведінкової моделі бджолиної колонії (ABC) - для структурної та параметричної ідентифікації інтервальних моделей; методи нейромережевої оцінки пози людини (HPE) на основі моделей YOLOv8-pose та MediaPipe Hand Landmarker – для детекції анатомічних точок у відеопотоці; методи програмної інженерії та розробки AR-застосунків - для проектування та реалізації комплексної програмної системи.

**Наукова новизна отриманих результатів.** У дисертаційній роботі отримано такі наукові результати:

*вперше* розроблено інтервальні математичні моделі динаміки відновлення кутів рухливості плечового суглоба з використанням адаптованих методів структурної та параметричної ідентифікації дискретних динамічних моделей, які дають змогу прогнозувати значення кутів на основі 3-4 попередніх сеансів терапії та забезпечити адаптивну корекцію терапевтичного протоколу, підвищуючи ефективність реабілітації шляхом раннього виявлення неоптимальної динаміки відновлення;

*вперше* запропоновано комплексну архітектуру медичної реабілітаційної програмної системи, яка, на відміну від існуючих, базується на оркестрації мікросервісів: аналізу трьох відеопотоків з пріоритизацією, прогнозування,

рекомендацій на основі генеративного штучного інтелекту та застосунку арт-терапії для гарнітури Magic Leap 2, а також вебзастосунку фізіотерапевта, що забезпечує зменшення тривалості повного циклу реабілітації в середньому на 12%;

*удосконалено* метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків, який базується на адаптивній фільтрації координат ключових анатомічних точок та алгоритмі геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху, що забезпечує зменшення проекційної похибки вимірювання кутів суглобів до клінічно прийняттого рівня та дає змогу автоматизувати об'єктивний контроль динаміки відновлення рухливості;

*набула подальшого розвитку* інформаційна технологія реабілітаційної арт-терапії, яка, на відміну від існуючих рішень, побудована на основі програмно-апаратних засобів з функціями автоматичної безмаркерної гоніометрії, інтервального прогнозування динаміки відновлення рухливості суглобів та генерації персоналізованого терапевтичного контенту, що забезпечує реалізацію повного циклу від автоматичного вимірювання до адаптивної корекції протоколу реабілітації верхніх кінцівок пацієнтів.

**Особистий внесок здобувача.** Основні наукові результати, що виносяться на захист, отримані автором самостійно. У працях, опублікованих у співавторстві, здобувачеві належить:

[21] – розробка програмної архітектури вебсистеми гоніометричного аналізу, реалізація модулів нейромережевої детекції анатомічних точок та обчислення кутів суглобів;

[45] – проектування та реалізація AR-застосунку арт-терапії для гарнітури Magic Leap 2, розробка модулів просторового позиціонування, тактильної компенсації та рендерингу текстур малювання;

[46] – розробка методу автоматичного гоніометричного аналізу на базі нейронних мереж, реалізація алгоритмів адаптивної фільтрації координат та геометричного злиття вимірювань від трьох камер;

[47] – розробка методу безмаркерного трикамерного вимірювання кутів суглобів, реалізація конвеєрної архітектури підсистеми відеоаналізу;

[48] – розробка комплексної архітектури AR-системи реабілітації, інтеграція компонентів у замкнутий адаптивний цикл прогнозування та корекції терапевтичного протоколу, проведення клінічного дослідження;

[49] – розробка рекомендаційної системи підтримки фізичної реабілітації на основі корекції терапевтичних зображень, реалізація багатоступеневого AI-контур генеративної арт-терапевтичного контенту.

**Апробація результатів дисертації.** Основні положення та результати дисертаційної роботи доповідались та обговорювались на таких наукових конференціях:

– 14th IEEE International Conference on Advanced Computer Information Technologies (ACIT-2024), Ceske Budejovice, Czech Republic, 2024;

– 15th IEEE International Conference on Advanced Computer Information Technologies (ACIT-2025), Sibenik, Croatia, 2025;

– школа-семінар молодих вчених і студентів «Комп'ютерні інформаційні технології» (CIT'2023, CIT'2025, CIT'2026), Тернопіль, ЗУНУ, 2023–2026 рр.

**Публікації.** За результатами дисертаційного дослідження опубліковано 9 наукових праць, зокрема 4 статті у періодичних фахових виданнях, з яких 1 стаття у закордонному періодичному виданні, що індексується наукометричною базою Scopus (Q2), та 3 у виданнях України категорії «Б», а також 5 публікацій у матеріалах конференцій, 2 з яких індексуються наукометричною базою Scopus.

**Структура та обсяг дисертації.** Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Повний обсяг дисертації становить 236 сторінок, з них 164 сторінок основного тексту. Робота містить 25 рисунків, 16 таблиць, список використаних джерел із 132 найменувань та 8 додатки.

**Декларація про використання ШІ.** Під час підготовки роботи автором використано Claude (Anthropic) для пошуку та первинного опрацювання літературних джерел, а також виявлення та виправлення граматичних,

орфографічних та стилістичних помилок. Після застосування цих інструментів автором проведено ретельний перегляд і внесено необхідні зміни, беручи на себе повну відповідальність за остаточний зміст дисертаційної роботи.

**Зв'язок роботи з науковими програмами, планами, темами.** Дисертаційна робота виконана на кафедрі комп'ютерних наук Західноукраїнського національного університету відповідно до плану науково-дослідних робіт. Здобувач брав участь у виконанні молодіжного держбюджетного дослідження «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (номер державної реєстрації 0124U000076), держбюджетного дослідження «Інформаційна технологія забезпечення ефективної акумуляції електроенергії гібридною електростанцією» (номер державної реєстрації 0126U002209) та молодіжного держбюджетного дослідження «Математичне та програмне забезпечення функціонування високоефективної гібридної когенераційної установки» (номер державної реєстрації 0126U002213).

**Практичне значення отриманих результатів** полягає у створенні комплексної програмної системи реабілітаційної арт-терапії на основі AR-технологій, яка може використовуватися у клінічній практиці для об'єктивного автоматичного контролю динаміки відновлення рухливості суглобів, прогнозування результатів реабілітації та адаптивної корекції терапевтичного протоколу.

Результати дисертаційної роботи впроваджено:

– у процес проведення реабілітаційних сеансів КНП «Тернопільська обласна клінічна лікарня» ТОО, де метод безмаркерної гоніометрії та програмне забезпечення системи AR-реабілітації використовуються для об'єктивного вимірювання кутів рухливості суглобів та адаптивної корекції терапевтичного протоколу, що забезпечило скорочення курсу реабілітації в середньому на 12% (акт про впровадження результатів дисертаційної роботи від 11 травня 2026 р.);

– у науково-дослідну роботу кафедри фізичної терапії, ерготерапії та фізичного виховання Тернопільського національного медичного університету імені І.Я. Горбачевського МОЗ України, де спосіб визначення кутів суглобової

рухливості та програмний комплекс для реабілітації верхніх кінцівок засобами арт-терапії в середовищі доповненої реальності використовуються при виконанні спільного науково-дослідного проекту з кафедрою комп'ютерних наук ЗУНУ (акт про впровадження результатів дисертаційної роботи від 15 травня 2026 р.);

– у виробничий процес Тернопільської філії компанії ТОВ «ЕЛЕКС», де результати дисертаційної роботи використовуються при розробці бізнес-рішень у сфері AR/VR-застосунків (довідка про впровадження від 11 травня 2026 р.);

– в освітній процес кафедри комп'ютерних наук факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету при підготовці студентів спеціальності F2 (121) – Інженерія програмного забезпечення, зокрема при викладанні дисциплін «Архітектура та проектування програмного забезпечення», «Інтелектуальний аналіз даних» та «Інтервальні обчислення» (акт про впровадження результатів дисертаційної роботи від 3 червня 2026 р.).

## РОЗДІЛ 1

### АНАЛІЗ МЕТОДІВ І ЗАСОБІВ АВТОМАТИЧНОГО ВИМІРЮВАННЯ КУТІВ СУГЛОБІВ ТА ПРОГНОЗУВАННЯ ДИНАМІКИ РЕАБІЛІТАЦІЇ

У цьому розділі проведено комплексний аналіз літературних джерел, пов'язаних із проблематикою автоматичного вимірювання кутів суглобів верхніх кінцівок, застосування технологій віртуальної та доповненої реальності у поєднанні з арт-терапією для фізичної реабілітації, прогнозування динаміки відновлення рухливості суглобів та підтримки клінічних рішень фізіотерапевта.

У підрозділі один проаналізовано сучасні системи фізичної реабілітації з використанням VR/AR-технологій та арт-терапії. Досліджено ефективність VR-реабілітації для верхніх кінцівок на основі систематичних оглядів та мета-аналізів, систематизовано парадигми інтеграції арт-терапії з технологіями імерсивного середовища та обґрунтовано переваги AR-арт-терапії для клінічної реабілітації. У підрозділі два проведено аналіз методів вимірювання кутів суглобів – від традиційних механічних гоніометрів до сучасних безмаркерних систем оцінки пози на основі глибокого навчання. Розглянуто обмеження систем безмаркерної гоніометрії на основі одного відеопотоку та визначено потребу у системах безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією. У підрозділі три проаналізовано методи побудови математичних моделей динаміки та прогнозування результатів реабілітації, зокрема інтервальні моделі на основі аналізу інтервальних даних, що дозволяють враховувати невизначеність вимірювань на коротких вибірках. У підрозділі чотири розглянуто системи підтримки клінічних рішень та рекомендаційні системи у фізичній реабілітації, обґрунтовано потребу в рекомендаційній системі, що реалізує рекомендації через корекцію терапевтичних зображень. У підрозділі п'ять на підставі проведеного аналізу сформульовано постановку задачі дисертаційного дослідження.

Результати цього розділу опубліковано автором у працях [45, 47, 48].

## **1.1. Аналіз систем вимірювання кутів суглобів та VR/AR-технологій у фізичній реабілітації**

Фізична реабілітація осіб із порушеннями рухових функцій верхніх кінцівок є одним із найбільш ресурсомістких та тривалих напрямків відновної медицини. Для України ця проблема набуває особливої гостроти: повномасштабне вторгнення 2022 року спричинило безпрецедентне зростання потреби у реабілітаційних послугах, зокрема для осіб із пораненнями кінцівок, а існуюча інфраструктура реабілітації залишається фрагментованою та недостатньою [51, 52]. Традиційні методи фізичної реабілітації, незважаючи на їхню клінічну обґрунтованість, часто стикаються з проблемою низької мотивації пацієнтів до тривалих курсів лікування та високим рівнем відмови від терапії, особливо при довгостроковій реабілітації верхніх кінцівок після інсульту чи травм [53].

Технології віртуальної реальності (VR) та доповненої реальності (AR) суттєво змінюють підходи до фізичної реабілітації. Систематичний Кокранівський огляд Laver та ін. [23], що охопив 72 дослідження із загальною кількістю 2470 учасників, та його оновлена версія 2025 року [54] встановили, що VR-реабілітація статистично значимо покращує функцію верхніх кінцівок та повсякденну активність порівняно з конвенційною терапією. Maier та ін. [25] у рандомізованому контрольованому дослідженні показали, що специфічне VR-тренування (на відміну від неспецифічного) забезпечує кращі результати моторного відновлення після інсульту. Систематичний огляд та мета-аналіз Karatians та ін. [26] підтвердив ефективність VR-інтервенцій для реабілітації верхніх кінцівок, зазначивши, що ігрові VR-системи демонструють найбільш виражений ефект на моторну функцію.

Мета-аналіз Peng та ін. [27] Soleimani та ін. [55] виявили, що VR-тренування має позитивний вплив на функцію верхніх кінцівок у пацієнтів з підгострим та хронічним інсультом. Mekbib та ін. [28] у мета-аналізі 17 рандомізованих контрольованих досліджень встановили, що VR-терапія забезпечує значиме покращення рухових показників верхніх кінцівок. Bortone та ін. [56] продемонстрували ефективність поєднання носимих гаптичних пристроїв з

імерсивною VR для реабілітації дітей з нейромоторними порушеннями, що підкреслює широкий спектр застосування VR-підходів для різних вікових груп та діагнозів.

Доповнена реальність (AR), на відміну від повністю імерсивної VR, накладає цифрові елементи на реальне середовище, зберігаючи зв'язок пацієнта з фізичним оточенням. Систематичний огляд та мета-аналіз застосування AR у фізичній терапії, проведений Vinolo Gil та ін. [57], виявив, що AR у поєднанні з конвенційною терапією використовується для покращення фізичної працездатності, лікування порушень балансу та функціональності верхніх і нижніх кінцівок у пацієнтів після інсульту. Водночас, незважаючи на позитивні результати в окремих дослідженнях, відсутній консенсус щодо оптимальної AR-системи для кожної клінічної області через значну різноманітність інтервенцій та вимірюваних параметрів. Нещодавній мережевий мета-аналіз Zhang та ін. [58] порівняв ефекти VR із різними модальностями на відновлення верхніх кінцівок після інсульту і встановив, що неімерсивні ігрові консолі виявились найефективнішою інтервенцією для покращення моторної функції, тоді як імерсивні VR-гарнітури не продемонстрували значних переваг над конвенційною терапією, що підкреслює важливість правильного вибору типу імерсії та модальності взаємодії.

Гейміфікація та мотиваційні аспекти VR/AR-реабілітації значно впливають на прихильність пацієнтів до лікування. Концепція «стану потоку» (flow state), запропонована Чиксентміхалї, описує оптимальний психологічний стан, коли пацієнт повністю поглинений діяльністю, що з'являється при правильному балансі між складністю завдання та рівнем навичок [59]. VR-ігри для реабілітації спеціально розроблені для підтримання цього стану, автоматично скоригувавши складність вправ на основі успішності пацієнта. Платформи RAPAEL (від компанії Neofect), Turomotion VirtualRehab та інші демонструють позитивні результати у клінічних дослідженнях [60, 61]. Бар'єри до впровадження VR у реабілітацію, систематизовані Glegg та Levas [62], включають високу вартість обладнання, відсутність стандартизованих протоколів та необхідність навчання персоналу, що підкреслює потребу у розробці доступних та клінічно інтегрованих рішень [63].

Арт-терапія – це форма терапевтичного втручання, яка використовує творчі процеси (малювання, скульптуру, колаж) для покращення фізичного, психічного та емоційного благополуччя пацієнтів. У контексті фізичної реабілітації арт-терапія набуває особливого значення завдяки подвійному терапевтичному впливу: по-перше, сам процес малювання або ліплення потребує координованих рухів верхніх кінцівок, що забезпечує моторне тренування у функціонально значимому контексті; по-друге, творча діяльність активує мотиваційні центри мозку, знижує рівень кортизолу та покращує психоемоційний стан [29, 30]. Дослідження Kaimal та ін. продемонструвало, що 45 хвилин арт-терапевтичної діяльності призводять до статистично значимого зниження рівня кортизолу – гормону стресу – незалежно від попереднього досвіду пацієнта в образотворчому мистецтві, що є вагомим аргументом для інтеграції арт-терапії у реабілітаційні програми.

Систематичний огляд та мета-аналіз активної візуальної арт-терапії, проведений Joschko та ін. [64] і опублікований у JAMA Network Open, охопив 69 рандомізованих клінічних досліджень із загальною кількістю 4200 учасників. Результати показали, що активна візуальна арт-терапія асоціюється з терапевтичними перевагами для низки клінічних показників, зокрема зниження тривожності, покращення якості життя та емоційного стану. Систематичний огляд Lo та ін. [31] ідентифікував позитивний вплив креативних арт-терапій на щоденну активність, моторну функцію кінцівок, дрібну моторику та емоційне благополуччя пацієнтів після інсульту, хоча підкреслив необхідність подальших рандомізованих контрольованих досліджень для формування доказової бази.

Рандомізоване контрольоване дослідження Kongkasuwan та ін. [32] показало, що включення творчої арт-терапії (малювання та ліплення) до стандартної реабілітаційної програми статистично значимо покращує функціональну незалежність пацієнтів після інсульту порівняно з контрольною групою, яка отримувала лише конвенційну терапію. Скопінговий огляд Brown та ін. [65], опублікований у Physical Therapy, проаналізував 76 досліджень і встановив, що включення мистецтва в реабілітацію покращує результати у психомоторному, когнітивному та афективному доменах. Музична терапія та танцювальна терапія

були найбільш дослідженими формами, однак візуальна арт-терапія (малювання, скульптура) демонструвала особливий потенціал для моторної реабілітації верхніх кінцівок завдяки безпосередньому залученню дрібної та великої моторики рук.

Окремий напрямок досліджень фокусується на арт-терапії, специфічно спрямованій на відновлення верхніх кінцівок. Christiansen та ін. [66] розробили програму Art skill-based Rehabilitation Training (ART) – новий підхід до моторного тренування верхніх кінцівок, що базується на виконанні завдань із малювання та трасування. Програма передбачала дев'ять одногодинних сесій малювання обома руками для пацієнтів після інсульту в умовах стаціонару. Дослідження здійсненності продемонструвало високу прийнятність програми: із 38 залучених пацієнтів 32 завершили курс (утримання 84%), а учасники повідомили про високий рівень задоволення та мотивації до продовження. Важливо, що навчені непрофесіонали були здатні безпечно та ефективно проводити реабілітаційні сесії, що свідчить про потенціал масштабування такого підходу у клінічних умовах з обмеженими ресурсами.

Інтеграція арт-терапії з технологіями віртуальної реальності відкриває принципово нові можливості для реабілітації. Iosa та ін. [67] описали так званий «ефект Мікеланджело» – феномен, при якому взаємодія з художніми стимулами у VR-середовищі покращує моторні показники пацієнтів. У дослідженні за участю 20 здорових добровольців та 4 пацієнтів після інсульту було показано, що під час VR-завдання, де учасники мали ілюзію малювання копій шедеврів класичного мистецтва, вони демонстрували коротші траєкторії рухів та нижче суб'єктивне сприйняття фізичного навантаження порівняно з контрольним завданням із простим зафарбовуванням полотна. Цей ефект пояснюється нейроестетичною стимуляцією: взаємодія з мистецтвом нейробіологічно індукує мотивацію та афективне збудження, що є фундаментальними аспектами нейрореабілітації.

Подальший розвиток цього напрямку представлений у рандомізованому контрольованому дослідженні Tieri та ін. [44], де 40 пацієнтів після інсульту були рандомізовані у дві групи: експериментальну (VR-арт-терапія) та контрольну (конвенційна фізична терапія верхніх кінцівок). Протокол включав заміну трьох

щотижневих сесій конвенційної терапії на сесії VR-арт-терапії з використанням «ефекту Мікеланджело» протягом 4 тижнів. Після 12 сесій пацієнти експериментальної групи продемонстрували статистично значиме покращення незалежності у повсякденній діяльності, м'язової сили верхніх кінцівок та зменшення спастичності порівняно з контрольною групою. Автори підкреслюють, що чим більшою була залученість пацієнта у терапію, тим кращими були отримані результати, що підтверджує мотиваційну роль арт-компоненту.

Розвиток комерційних VR-додатків для тривимірного малювання – Google Tilt Brush, Gravity Sketch, Quill – відкрив нові перспективи для арт-терапевтичної практики у реабілітації. Дослідження Tap та ін. [68] показало, що малювання у VR-середовищі (Tilt Brush) має потенціал для зниження тривожності, при цьому тривимірне малювання забезпечує унікальний терапевтичний досвід, недоступний при класичному двовимірному малюванні: пацієнт може створювати об'ємні об'єкти, обходити їх з усіх боків та навіть перебувати всередині власного твору. Скопінговий огляд Наджіранаяї та ін. [69] систематизував дослідження арт-терапії у VR та виявив три основні категорії: танцювальну терапію, музичну терапію та візуальну арт-терапію, кожна з яких продемонструвала потенціал для нейрореабілітації при різних неврологічних станах – інсульті, хворобі Паркінсона, набутих ушкодженнях головного мозку та деменції.

Особливо перспективним є застосування VR для арт-терапії у пацієнтів з обмеженою моторикою. VR-середовище дозволяє адаптувати масштаб рухів: мінімальні жести пацієнта у фізичному світі можуть трансформуватися у широкі мазки пензля у віртуальному просторі, що дає змогу пацієнтам з обмеженим діапазоном рухів виконувати повноцінну арт-терапевтичну діяльність [69]. Дослідження Shojaei та ін. [70] вивчило перспективи застосування генеративного штучного інтелекту в арт-терапії та виявило, що AI-генеровані зображення можуть слугувати стимулом для творчої активності, знижуючи бар'єр входу для пацієнтів, які не мають художніх навичок. Адаптивний підхід, де складність арт-терапевтичного завдання автоматично коригується відповідно до поточних

можливостей пацієнта, є ключовим принципом сучасних технологічно опосередкованих реабілітаційних систем [113].

На відміну від повністю імерсивних VR-систем, доповнена реальність (AR) зберігає зв'язок пацієнта з фізичним оточенням, накладаючи цифрові елементи на реальне середовище. Це є критично важливим для фізичної реабілітації, оскільки пацієнт продовжує бачити власне тіло, реальний простір та фізіотерапевта, що підвищує безпеку та дозволяє здійснювати клінічний нагляд у реальному часі. AR-арт-терапія поєднує ці переваги з мотиваційним потенціалом творчої діяльності: пацієнт малює у доповненій реальності, використовуючи жести власних рук, що одночасно забезпечує моторне тренування та творчу самореалізацію. Прототип AR-системи арт-терапії для гарнітури Magic Leap 2, розроблений авторами [45], продемонстрував технічну здійсненність такого підходу та отримав позитивні відгуки від фізіотерапевтів під час попередньої апробації.

Систематизація існуючих підходів дозволяє виділити три основні парадигми інтеграції арт-терапії у технологічно опосередковану реабілітацію. Перша парадигма – пасивна VR-арт-терапія – передбачає споглядання та імітацію відтворення художніх творів у VR для нейроестетичної стимуляції (ефект Мікеланджело). Друга парадигма – активна VR-арт-терапія – включає безпосереднє створення арт-об'єктів у тривимірному VR-просторі за допомогою спеціалізованих додатків (Tilt Brush, Quill). Третя парадигма – AR-арт-терапія – поєднує творчу діяльність із збереженням зв'язку з реальним середовищем та фізичними рухами верхніх кінцівок. Саме третя парадигма є найбільш перспективною для клінічної реабілітації верхніх кінцівок, оскільки вона забезпечує одночасно моторне тренування (реальні рухи руками), творчу мотивацію (арт-терапевтичне завдання) та безпеку (пацієнт бачить реальне оточення та фізіотерапевта).

Аналіз літератури виявляє істотний розрив між наявними дослідженнями та потребами клінічної практики. Існуючі VR-арт-терапевтичні системи, побудовані на «ефекті Мікеланджело», демонструють клінічну ефективність, але обмежені повністю імерсивним характером та не передбачають інтеграції з об'єктивними

засобами оцінки моторної функції. Комерційні VR-додатки для малювання (Tilt Brush) не адаптовані для клінічного використання та не забезпечують зворотного зв'язку щодо терапевтичного прогресу. AR-системи для реабілітації зазвичай фокусуються на ігрових завданнях без арт-терапевтичної компоненти. Таким чином, комплексна система, яка інтегрує AR-арт-терапію з автоматичним контролем рухових показників, математичним прогнозуванням динаміки відновлення та адаптивним управлінням терапевтичним процесом, відсутня у науковій літературі, що підтверджує актуальність дисертаційного дослідження.

## **1.2. Аналіз методів вимірювання кутів суглобів та безмаркерних систем оцінки пози**

Об'єктивне кількісне вимірювання діапазонів рухів у суглобах є фундаментальною складовою процесу реабілітації верхніх кінцівок, оскільки саме динаміка зміни ROM визначає ефективність терапевтичних втручань та дозволяє коригувати план лікування [3]. Традиційне вимірювання кутів суглобів за допомогою механічного гоніометра залишається широко використовуваним методом у клінічній практиці. Однак цей метод має суттєві обмеження: помилка вимірювання складає від 5 до 10 градусів, залежно від навичок фізіотерапевта, особливостей анатомії пацієнта та якості калібрування приладу [3, 4]. Дослідження Nancosk та ін. [72] виявило, що між-спостерігальна надійність (inter-rater reliability) при вимірюванні колінного суглоба гоніометром характеризується коефіцієнтом ICC = 0,65-0,84, що відповідає стандартній похибці 8-14°. Така варіабельність означає, що зміна показника ROM менше ніж на 15° між двома спостерігачами не може бути інтерпретована як реальна динаміка відновлення. Крім того, мануальна гоніометрія дозволяє фіксувати лише статичні крайні положення суглоба, не забезпечуючи безперервного вимірювання під час виконання руху.

Цифрові інклінометри та програмні додатки на основі смартфонів пропонують певне поліпшення, дозволяючи зберігати виміри та відстежувати їх у часі [5], однак точність таких пристроїв залишається обмеженою через залежність від якості датчиків та корекції орієнтації девайса. Системи на основі інерціальних вимірювальних модулів (IMU) – таких як Werium та MPU 6050 – дозволяють

отримувати дані про орієнтацію суглобів у реальному часі [6, 7], однак накопичення помилки (drift) при тривалому використанні та необхідність калібрування для кожного індивідуального пацієнта являють собою серйозні практичні проблеми. Для клінічної реабілітації, де сеанси тривають 30-60 хвилин, дрейф ІМУ-датчиків може досягати 5-10° за годину, що суттєво знижує достовірність вимірювань у другій половині сеансу.

Маркерні системи фіксування руху (Vicon, OptiTrack, Qualisys) вважаються золотим стандартом у науковій практиці, забезпечуючи субміліметрову точність визначення координат при частоті до 2000 Гц та можливість відстежувати до сотень маркерів одночасно [9, 73]. Дослідження Merriault та ін. [73] підтвердило високу точність системи Vicon із середньою похибкою позиціонування менше 0,5 мм. Проте використання маркерних систем у клінічних умовах та реабілітаційних центрах обмежене через надзвичайно високу вартість обладнання (50 000-150 000 USD), необхідність спеціалізованого простору з контрольованим освітленням та тривалу підготовку пацієнта – встановлення маркерів займає 20-40 хвилин, що є неприйнятним для щоденної клінічної практики [10]. Узагальнюючи, контактні методи вимірювання (механічні гоніометри, інклінометри) мають нижчу точність та трудомісткі у використанні, маркерні системи забезпечують найвищу точність, але недоступні для більшості клінік, а ІМУ-датчики страждають від дрейфу. Тому глобальна потреба в розробленні доступних та водночас точних методів автоматичного вимірювання кутів суглобів залишається гострою.

Розвиток методів глибокого навчання для оцінювання пози людини (Human Pose Estimation, HPE) відкрив принципову можливість безмаркерного аналізу руху на стандартних камерах без спеціалізованого обладнання [9, 12, 74]. Систематичний огляд Lam та ін. [75], опублікований у Journal of NeuroEngineering and Rehabilitation, проаналізував 65 досліджень застосування безмаркерних систем захоплення руху (Markerless Motion Capture, MMC) для клінічних вимірювань у реабілітації та встановив, що MMC-технології найчастіше використовуються для ідентифікації симптомів, виявлення відмінностей у патернах руху між пацієнтами та здоровими особами, а також для оцінки ефективності терапевтичних

інтервенцій. Огляд виявив зростаючий інтерес до застосування НРЕ в клінічних умовах, однак підкреслив необхідність подальшої валідації для досягнення клінічної прийнятності [75].

Системи НРЕ на основі глибокого навчання аналізують відеопотік за допомогою нейронних мереж та виділяють координати ключових точок тіла (суглобів, опорних точок). OpenPose, розроблена в Carnegie Mellon University, була першою системою НРЕ, яка забезпечила відстежування 25 ключових точок на 2D-зображенні у реальному часі [12]. Архітектура використовує Part Affinity Fields для одночасного виявлення та асоціації ключових точок кількох осіб, проте виділяє переважно точки туловища та кінцівок без детальної деталізації кистей рук. BlazePose, розроблена Google [13], спеціально оптимізована для мобільних пристроїв і забезпечує швидку обробку за рахунок зменшеного розміру моделі, що робить її привабливою для розгортання на периферійних пристроях з обмеженими обчислювальними ресурсами.

MediaPipe Pose Landmarker забезпечує виділення 33 ключових точок (COCO33), включаючи детальні позначки рук та обличчя, що є критично важливим для клінічних додатків реабілітації [14, 76]. Це дозволяє вимірювати кути не лише великих суглобів (плечовий, ліктьовий, тазостегновий), але й дрібніших (зап'ястя, пальці). MediaPipe Hand Landmarker надає 21 точку для кожної кисті, а MediaPipe Holistic об'єднує цілісний скелет із детальними моделями кистей, що дозволяє будувати комплексні моделі верхніх кінцівок з високою точністю локалізації.

Огляд Roggio та ін. [77], що систематизував моделі оцінки пози для аналізу руху та постави, встановив, що MediaPipe є однією з небагатьох моделей, для яких проведено кількісну валідацію проти маркерних систем (Qualisys, Vicon), що підтвердила високу точність вимірювань – сильна кореляція з еталоном ( $r = 0,91 \pm 0,08$ ) для рухів верхніх кінцівок та добра до відмінної узгодженість просторово-часових параметрів ( $ICC > 0,75$ ), тоді як для PoseNet, HRNet та EfficientPose такі порівняння із золотим стандартом на момент огляду відсутні.

Альтернативним підходом є MoveNet (Google), який пропонує дві варіанти: Lightning (швидка, 17 ключових точок) та Thunder (точніша, 17 точок) [15]. На

відміну від MediaPipe з 33 точками, MoveNet має менше суглобів та не включає деталізацію рук, що зменшує його застосовність для вимірювання ROM верхніх кінцівок. ViTPose та інші transformer-based моделі НРЕ обіцяють кращу точність завдяки глобальним механізмам уваги, але вимагають значно більших обчислювальних ресурсів [78, 79], що ускладнює їх розгортання в умовах реального часу на клінічному обладнанні.

Окрему категорію складають YOLO-сімейство детекторів пози (YOLOv8-pose, YOLOv11-pose) від Ultralytics, які поєднують задачу виявлення людини та регресії 17 ключових точок COCO у єдиному прямому проході згорткової мережі [17], зокрема Dong та Du [80] запропонували модифікацію з модулем контекстної координатної уваги (CCAM), що забезпечує покращення точності на 3,5% AP порівняно з базовою моделлю. На відміну від MediaPipe, де детекція побудована як двоетапний конвеєр (person detector → landmark regressor), YOLO-архітектура забезпечує нижчу обчислювальну вартість (9,2 GFLOPs для варіанту nano проти 30+ GFLOPs для MediaPipe Pose full) і, що важливіше, підтримує експорт у формати ONNX та OpenVINO IR, що дозволяє отримати додаткове прискорення у 1,3-1,8 раза на CPU без GPU за рахунок векторизованих інструкцій AVX-512 та FP16-квантизації [81]. Компромісом є менша кількість ключових точок (17 замість 33) та відсутність вбудованої деталізації кистей, що компенсується гібридним поєднанням YOLOv8-pose для тіла з окремим викликом MediaPipe Hand Landmarker для зап'ястних кутів. Дослідження валідації, які порівнюють безмаркерні системи з золотим стандартом (Vicon), показують, що MediaPipe досягає клінічно прийнятної точності для основних суглобів (плечовий, ліктьовий), але має обмеження для зап'ястя та дрібних суглобів [18]. Концепція мінімальної клінічно значимої різниці (MCID) для ROM зазвичай становить 5°-15° залежно від суглоба та типу патології [72], що визначає вимогу до точності системи - не гірше 5° для основних суглобів верхньої кінцівки.

Порівняльну характеристику розглянутих моделей НРЕ за критеріями кількості ключових точок, швидкодії, підтримки 3D та придатності для вимірювання ROM верхніх кінцівок наведено в таблиці 1.1.

Таблиця 1.1.

Порівняння моделей оцінки пози людини (HPE) для задач безмаркерної  
гоніометрії

Модель	Ключових точок	Архітектура	Швидкість (FPS)	3D-підтримка	Руки/пальці	Придатність для ROM верхніх кінцівок
OpenPose [12]	25	Bottom-up CNN	~15	Ні	Окремий модуль	Обмежена (мало точок рук)
BlazePose / MediaPipe [14, 76]	33	Top-down, lightweight	~30	Pseudo-3D	Так	Висока (детальні руки, швидкість)
MoveNet [15]	17	Thunder / Lightning	~30-50	Ні	Ні	Низька (відсутні точки рук)
ViTPose [16]	17 / 133	Vision Transformer	~10-20	Ні	Так (133)	Середня (повільна, немає 3D)
YOLOv8-pose [17]	17	Single-shot CNN	~30-60	Ні	Ні	Висока (швидкість, OpenVINO)

Порівняння моделей HPE (табл. 1.1) демонструє, що для задач безмаркерної гоніометрії верхніх кінцівок кожна з моделей має свої переваги та обмеження. BlazePose / MediaPipe Pose Landmarker забезпечує найбільшу деталізацію анатомічних точок (33 точки з вбудованою детекцією кистей), проте має вищі обчислювальні вимоги. YOLOv8-pose від Ultralytics поєднує високу швидкодію (до 60 FPS) з низькою обчислювальною вартістю (9,2 GFLOPs для варіанту nano) та підтримкою експорту в OpenVINO для прискорення на CPU без GPU, що робить її оптимальною для розгортання на стандартному клінічному обладнанні. Водночас YOLOv8-pose визначає лише 17 ключових точок COCO (Common Objects in

Context) без вбудованої деталізації кистей, що є недостатнім для точного вимірювання зап'ястних кутів. Тому для оптимізованої реалізації методу безмаркерної гоніометрії (підрозділ 2.1) обрано гібридну архітектуру, що поєднує YOLOv8-pose як основний детектор тіла та MediaPipe Hand Landmarker для детекції 21 точки кожної кисті. Саме таку гібридну архітектуру обрано для оптимізованої реалізації методу безмаркерної гоніометрії (підрозділ 2.1).

Клінічна валідація безмаркерних систем є критичним етапом для їх впровадження у реабілітаційну практику. Wade та ін. [18] у систематичному огляді поточних безмаркерних методів для клінічної біомеханіки ходи проаналізували обмеження існуючих підходів та визначили, що основними бар'єрами до клінічного впровадження є недостатня валідація на реальних пацієнтах, обмежена точність у нестандартних позах та чутливість до умов освітлення. D'Haene та ін. [82] підтвердили надійність 3D безмаркерної системи захоплення руху для кількісного аналізу ходи, отримавши високу узгодженість із золотим стандартом ( $LCC > 0,96$ ) та міжсесійну надійність  $RMSE < 3^\circ$ . Пілотне дослідження валідності та надійності безмаркерного вимірювання кінематики верхніх кінцівок, проведене Lam та ін. [83], порівняло ММС-систему з конвенційною гоніометрією та встановило, що безмаркерна система може ефективно використовуватись для внутрішньо-суб'єктних вимірювань у домашній та телереабілітаційній практиці завдяки хорошій надійності, низькій вартості та портативності, хоча для повної заміни клінічної гоніометрії потрібна додаткова калібрація.

Для вимірювання кутів суглобів саме плечового поясу, що є предметом даного дослідження, Pottorf та ін. [84] продемонстрували надійність та конкурентну валідність безмаркерної 3D-системи із однією камерою для оцінки гленогумеральної рухливості: системні кореляції з гоніометрією для згинання, відведення, зовнішньої та внутрішньої ротації плеча виявились сильними ( $r > 0,85$ ), що підтверджує придатність безмаркерних технологій для клінічної оцінки ROM плечового суглоба. Дослідження точності AI-базованого визначення кутів для реабілітації плеча та ліктя, проведене Klein та ін. [85], порівняло різні фреймворки HPE і встановило, що середня абсолютна похибка вимірювання кутів плечового та

ліктьового суглобів складає  $5-17^\circ$  залежно від моделі та типу руху, при цьому оптимальні результати досягаються при фронтальному та сагітальному ракурсах камери.

Критичною проблемою безмаркерних систем із однією камерою є проекційна похибка – спотворення вимірюного кута при невдалому ракурсі камери відносно площини руху суглоба. Yeung та ін. [86] продемонстрували вплив кутів огляду камери на точність відстежування кінематичних патернів ходи за допомогою Azure Kinect та Kinect v2: похибка суттєво зростає при відхиленні оптичної осі камери від площини руху, досягаючи  $30-50^\circ$  при перпендикулярному розташуванні. Albert та ін. [87] при порівнянні Azure Kinect та Kinect v2 з еталонною системою Vicon підтвердили, що безмаркерні системи потребують оптимального позиціонування камери для досягнення клінічно прийнятної точності. Ця залежність від ракурсу є фундаментальним обмеженням систем із однією камерою і не може бути усунена програмними засобами в рамках одного відеопотоку [88].

Багатокамерні підходи розв'язують проблему проекційної похибки шляхом об'єднання вимірювань від кількох камер, розташованих під різними кутами. Hartley та Zisserman [88] описали математичні основи багатокамерної геометрії та методи тріангуляції для реконструкції 3D-координат із множинних 2D-проекцій. Slembrouck та ін. дослідили багатокамерну 3D-реконструкцію пози на основі скелетів OpenPose та продемонстрували суттєве зменшення похибки локалізації при об'єднанні даних від 3-4 камер [78]. Нещодавній огляд Nogueira та ін. [89] систематизував методи багатокамерної 3D-оцінки пози людини та виділив основні підходи: тріангуляцію, об'ємну реконструкцію та нейромережеве злиття, зазначивши, що тріангуляційні методи забезпечують оптимальний баланс між точністю та обчислювальною складністю для клінічних додатків реального часу.

Головною проблемою НРЕ залишається присутність шуму у виділених координатах: навіть авторитетні системи типу MediaPipe та YOLOv8-pose можуть видавати стрибки координат між кадрами, які не відповідають реальним рухам пацієнта, що спричиняє помилки при розрахунку кутів суглобів. Традиційні методи фільтрування включають ковзаючу середню, експоненціальне згладжування та

фільтр Калмана [19, 20], причому фільтр Калмана є найпопулярнішим методом у системах реального часу, оскільки він рекурсивний, ефективний з точки зору обчислень та може адаптуватися до динаміки процесу. Адаптивні варіанти фільтра Калмана, де параметри шуму вимірювань змінюються залежно від характеристик вхідного сигналу, забезпечують кращий компроміс між згладжуванням та швидкістю реакції порівняно з класичним фільтром із фіксованими параметрами [33, 112, 19].

1€ Filter (One Euro Filter), запропонований Casiez, Roussel і Vogel [20], є принципово іншою альтернативою фільтру Калмана: замість вектора стану і матричних операцій він використовує скалярне експоненційне згладжування із адаптивно-керованою частотою зрізу, що динамічно зростає пропорційно оцінці швидкості сигналу. Обчислювальна складність OneEuro становить  $O(1)$  на одну координату проти  $O(n^3)$  для Калмана при інверсії коваріаційної матриці, що робить його особливо привабливим для задач real-time обробки з обмеженими обчислювальними ресурсами - зокрема для багатопотокових конвеєрів, де фільтрація виконується незалежно для сотень координат ключових точок. Компромісом OneEuro порівняно з адаптивним Калманом є відсутність явної оцінки невизначеності (коваріаційної матриці) і відсутність явної кінематичної моделі. Водночас для задач гоніометрії у процесі реабілітації ці обмеження є несуттєвими: клінічна достовірність вимірювання ROM визначається узгодженістю з апаратно-валідованими еталонами, а не коваріаційною оцінкою; кінематична модель верхньої кінцівки у реабілітаційних вправах (повільні контрольовані рухи з частотою менше 1-2 Гц) не потребує складнішої моделі за константну швидкість, що імпліцитно закладена в адаптивну частоту зрізу OneEuro. Таким чином, у сучасній літературі представлено два різних за обчислювальними властивостями підходи до адаптивної фільтрації координат НРЕ - адаптивний фільтр Калмана з науково обґрунтованою оцінкою невизначеності та OneEuro-фільтр зі спрощеною обчислювальною моделлю, - кожен з яких має області переважного застосування залежно від апаратних обмежень цільової платформи.

Датчики глибини (depth sensors) являють собою проміжний підхід між маркерними та повністю безмаркерними системами: Azure Kinect, Intel RealSense та аналогічні датчики використовують структурне світло або часо-імпульсну топографію (ToF) для отримання 3D-карти сцени [86, 87]. Azure Kinect Body Tracking SDK забезпечує відстежування 32 суглобів скелету в реальному часі, що дозволяє отримувати кількісні дані про положення суглобів без маркерів. Однак ці системи залежать від наявності спеціалізованого обладнання, мають обмежену робочу відстань (0,5-3,5 м для Azure Kinect), чутливі до прямого сонячного світла і не є універсальним рішенням для клініки з обмеженим бюджетом. Крім того, компанія Microsoft припинила виробництво Azure Kinect DK у 2023 році, що ставить під сумнів довгострокову підтримку рішень на цій платформі.

Таким чином, аналіз існуючих методів вимірювання кутів суглобів виявляє чітку дихотомію. Систематизовані результати порівняння наведено в табл. 1.2. Контактні методи (гоніометри, інклінометри, IMU) є доступними, але мають обмежену точність та трудомісткі у використанні; маркерні системи (Vicon) забезпечують найвищу точність, але недоступні для більшості клінік; безмаркерні системи на базі НРЕ (MediaPipe, OpenPose, YOLO-pose) є доступними та не потребують контакту з пацієнтом, проте мають проблеми з шумом нейромережевої детекції та проекційною похибкою при конфігурації із однією камерою. Підхід із використанням декількох камер та адаптивною фільтрацією потенційно усуває ці обмеження, забезпечуючи точність на рівні 3-5° для основних суглобів, що відповідає клінічним вимогам. При цьому вибір технологічного стеку реалізації системи безмаркерної гоніометрії залежить від апаратних характеристик цільової платформи: для робочих станцій з достатнім обчислювальним ресурсом (8+ ядер CPU) доцільним є використання MediaPipe Pose Landmarker у поєднанні з адаптивним фільтром Калмана, що забезпечує максимальну деталізацію анатомічних точок (33 точки тіла + 21 × 2 точки кистей) та науково обґрунтовану фільтрацію з явною оцінкою невизначеності; для стандартних клінічних станцій без дискретної графіки доцільним є використання YOLO-pose у поєднанні з OneEuro-фільтром та субсемплінгом, що знижує обчислювальні вимоги у 1,5-2 рази

при збереженні клінічно прийнятної точності. Обидва варіанти реалізації, їх математичне обґрунтування та порівняльна оцінка наведені у підрозділі 2.1.

Таблиця 1.2.

## Порівняльний аналіз методів вимірювання кутів суглобів

Метод вимірювання	Точність	Вартість	Портативність	Автоматизація	Доступність для клінік	Основні обмеження
Механічний гоніометр [3, 4]	$\pm 5^\circ$	Низька	Висока	Ні	Висока	Суб'єктивність, міжоператорна варіабельність $\leq 10^\circ$
Смартфон-інклінометр [5]	$\pm 5-8^\circ$	Низька	Висока	Частково	Висока	Залежність від якості сенсорів пристрою
IMU-сенсори (Werium) [6, 7]	$\pm 2-3^\circ$	Середня	Середня	Так	Середня	Магнітний дрефт, необхідність калібрації, контактне кріплення
Маркерна MoCap (Vicon) [9, 73]	$< 1^\circ$	Дуже висока	Низька	Так	Низька	Лабораторні умови, висока вартість ( $> \$100K$ )
Depth-сенсори (Kinect) [86, 87]	$\pm 3-5^\circ$	Середня	Середня	Так	Середня	Оклюдії, знято з виробництва, обмежений діапазон
Безмаркерна 1 камера [10, 15]	$\pm 5-10^\circ$	Низька	Висока	Так	Висока	Проекційна похибка, залежність від ракурсу
<b>Безмаркерна 3 камери (запропонований метод)</b>	$3-5^\circ$	Низька	Висока	Так	Висока	Необхідність калібрації камер, залежність від освітлення

Як видно з таблиці 1.2, запропонований метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків поєднує клінічно прийнятну точність (3-5°, що відповідає внутрішньооператорній надійності механічного гоніометра) з низькою вартістю, високою портативністю та повною автоматизацією, що робить його оптимальним для інтеграції в систему AR-реабілітації.

### **1.3. Аналіз методів побудови моделей динаміки та прогнозування результатів реабілітації**

Однією із основних компонент програмної системи для реабілітації пацієнтів з порушеннями рухливості верхніх кінцівок є математичні моделі динаміки відновлення кутів рухливості суглобів. Основне завдання цих моделей – прогнозування процесів реабілітації на підставі початково проведеної серії сеансів, що дозволяє клініцистам адаптувати протокол терапії на основі індивідуальних траєкторій пацієнта, а не використовувати стандартні схеми [34]. Це особливо важливо у період раннього відновлення після інсульту, коли прогноз результату визначається в перші тижні реабілітації [64, 91]. Об'єктивний прогноз кількості сесій до досягнення нормативних значень рухливості суглобів дозволяє обґрунтовано формувати терапевтичні протоколи та своєчасно коригувати стратегію лікування. Для побудови математичних моделей динаміки відновлення використовуються результати, отримані в процесі реабілітації пацієнтів: часові ряди значень кутів згинання, розгинання та абдукції, зареєстровані під час кожного реабілітаційного сеансу. Ключовою вимогою до моделі є здатність прогнозувати наступні значення кутів на основі 3-4 попередніх вимірювань з урахуванням невизначеності, притаманної клінічним даним.

У теорії математичного моделювання задачу знаходження моделі, що адекватно відтворює поведінку об'єкта на основі експериментальних даних, називають задачею ідентифікації [92]. Ідентифікація охоплює як побудову структури моделі – вибір типу рівнянь, порядку динаміки, переліку змінних, – так і оцінювання її числових параметрів. У сучасній інженерії програмного забезпечення задача ідентифікації є ключовою для аналізу, керування та прогнозування поведінки технічних, медичних, екологічних та енергетичних

систем, де неможливо покладатися лише на аналітично отримані фізичні моделі [38]. Стандартний цикл ідентифікації включає кілька основних етапів: формулювання мети моделювання, збір даних, визначення загального вигляду моделі, оцінювання параметрів та валідація моделі. Зазначені етапи не є одноразовою лінійною послідовністю: у практиці вони виконуються ітеративно – невдала валідація моделі вимагає або зміни структури моделі, або повторної оцінки параметрів. У найзагальнішому вигляді розрізняють два взаємопов'язані різновиди ідентифікації: структурну та параметричну.

Регресійний аналіз є одним із найпоширеніших статистичних методів для моделювання динаміки процесів, де залежна змінна (кут рухливості суглоба) виражається як функція від незалежних змінних (час, попередні вимірювання, зовнішні фактори). У простому лінійному регресійному аналізі модель має вигляд залежності між вихідною характеристикою  $Y$  та вхідними змінними  $X$ , де параметри оцінюються за методом найменших квадратів (МНК). Для динамічних систем часто застосовується множинна регресія або логістична регресія, якщо дані мають нелінійний характер [35]. Переваги регресійного аналізу включають простоту інтерпретації результатів, оскільки коефіцієнти вказують на силу та напрямок впливу змінних. Однак регресійний аналіз має суттєві недоліки при моделюванні динамічних реабілітаційних процесів: припущення про лінійність залежностей не завжди відповідає реальним даним, автокореляція між послідовними спостереженнями порушує припущення про незалежність похибок, а метод чутливий до викидів та мультиколінеарності [35]. У задачах прогнозування динаміки реабілітації регресія погано справляється зі складними патернами, оскільки не враховує часові залежності без додаткових розширень. Крім того, при малих обсягах даних (10-20 сеансів) регресійний аналіз не забезпечує коректну оцінку статистичних характеристик вибірки.

Методи аналізу часових рядів, зокрема авторегресійні моделі, є спеціалізованими для моделювання динаміки процесів, де дані залежать від часу. Авторегресійний метод моделює поточне значення як лінійну комбінацію  $p$  попередніх значень, де  $p$  – порядок моделі. Розширення, такі як ARMA

(авторегресія з ковзним середнім) чи ARIMA (інтегрована ARMA), враховують тенденції, сезонність та стаціонарність [35]. Переваги авторегресійних методів включають простоту реалізації, високу точність для стаціонарних рядів та можливість врахування автокореляції. ARIMA добре справляється з нестаціонарними даними через диференціювання, що є корисним для моделювання поступових змін у динаміці відновлення. Недоліки авторегресійних методів полягають у припущенні лінійності, що обмежує їх застосування для нелінійних процесів з раптовими змінами [35, 93]. Найсуттєвішим обмеженням у контексті реабілітації є вимога достатньо довгих рядів спостережень: класичні моделі ARIMA потребують щонайменше 30-50 спостережень для надійної ідентифікації параметрів, тоді як у клінічних умовах кількість реабілітаційних сеансів для одного пацієнта зазвичай обмежена 10-20 точками. Це унеможливорює пряме застосування ARIMA для індивідуального прогнозування динаміки відновлення кутів суглобів.

Моделі на базі нейронних мереж, зокрема рекурентні (RNN), LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit), є потужними інструментами для моделювання динаміки часових рядів завдяки здатності захоплювати нелінійні залежності та довготривалі кореляції [94]. Систематичні огляди застосування машинного навчання для прогнозування моторних функцій у пацієнтів з інсультом демонструють перспективність цих підходів [94, 36]. Переваги нейронних мереж включають високу точність для нелінійних даних, автоматичне витягування ознак та адаптивність до великих датасетів. Проте нейромережеві моделі мають суттєві недоліки у контексті клінічного застосування: вони є «чорним ящиком» з низькою інтерпретованістю, що суперечить вимогам пояснюваності у медичних системах підтримки рішень [36, 95]; для навчання потрібні сотні-тисячі зразків, тоді як дані одного пацієнта складають лише 10-20 точок, що неминуче призводить до перенавчання; RNN/LSTM чутливі до гіперпараметрів, а їх навчання є обчислювально затратним порівняно з регресією.

Гаусівські процеси регресії (Gaussian Process Regression, GPR) пропонують імовірнісний підхід до прогнозування, забезпечуючи довірчі інтервали для кожного прогнозованого значення без необхідності дискретизації моделі [37]. GPR

визначає апостеріорний розподіл над функціями на основі ядрової функції коваріації та спостережуваних даних, що дозволяє природним чином оцінювати невизначеність прогнозу. Однак обчислювальна складність GPR становить  $O(n^3)$ , де  $n$  – кількість спостережень, що обмежує застосування при великих наборах даних. Крім того, вибір ядрової функції суттєво впливає на якість прогнозу, а для короткотривалих реабілітаційних програм (10-20 сеансів) GPR може давати надмірно широкі довірчі інтервали через брак даних для надійної оцінки гіперпараметрів ядра.

Таким чином, усі розглянуті методи мають спільні суттєві недоліки стосовно задачі прогнозування динаміки відновлення кутів рухливості суглобів: регресійні моделі та ARIMA не враховують невизначеність та похибки, притаманні даним реабілітаційних вимірювань; нейронні мережі хоча й здатні до апроксимації складних залежностей, вимагають значних обсягів даних, що є нереалістичним у клінічних умовах; GPR є обчислювально затратним. Дані, отримані з відеокамер або датчиків, містять похибки, пов'язані з неточністю вимірювань, неточністю алгоритмів детекції, девіацією рухової активності та випадковими рухами пацієнта. Ці похибки є обмеженими за значенням і можуть бути подані у вигляді інтервалів. Вибірки даних реабілітаційних сесій є короткими (10-20 точок), що унеможлиблює застосування класичних стохастичних методів, які потребують значних обсягів даних для оцінки статистичних характеристик розподілів [38].

Альтернативним підходом, що враховує зазначені обмеження, є методи аналізу інтервальних даних [38, 103]. Теоретичні основи інтервального аналізу було закладено у працях Moore [96], який формалізував арифметику інтервалів та показав, що будь-яка неперервна функція дійсних змінних може бути коректно розширена на інтервальні аргументи із збереженням гарантії включення істинного значення в обчислений інтервал. Подальший розвиток прикладних аспектів інтервального аналізу представлено у роботі Jaulin та ін. [97], де систематизовано методи інтервальних обчислень для задач оцінювання параметрів, верифікації обмежень та гарантованого розв'язування систем рівнянь. У контексті реабілітації результати вимірювань кутів з фіксованою абсолютною похибкою подають у

вигляді числових інтервалів, а аналіз цих даних проводять процедурами інтервального аналізу. Якщо похибка вимірювання кута становить  $\pm\Delta$ , то вимірне значення  $z$  представляється інтервалом  $[z - \Delta; z + \Delta]$ , що природно враховує невизначеність без необхідності припускати конкретний закон розподілу похибки. Такий підхід є особливо релевантним для безмаркерних систем гоніометрії, де похибка нейромережевого детектора є обмеженою, але не підпорядковується гаусівському розподілу.

Математичну модель динаміки відновлення кутів рухливості суглобів розглядають як дискретне (різницеве) рівняння, де модельоване значення кута  $v$  для  $k$ -го сеансу визначається через набір базисних функцій  $f_1, f_2, \dots, f_n$  від попередніх значень кутів та вектор параметрів  $g$ , значення яких необхідно оцінити на основі інтервальних даних [38, 104]. Рекурентний характер моделі дозволяє прогнозувати наступні значення кутів на основі  $d$  попередніх вимірювань (зазвичай  $d = 3-4$ ), що відповідає реальному клінічному сценарію. Якщо початкові умови для різницевої схеми задати у вигляді інтервалів, отриманих з експериментальних вимірювань з урахуванням похибок, то математична модель набуває форми інтервального різницевого рівняння. Спираючись на гіпотезу, що обчислені інтервальні оцінки модельованої характеристики мають належати отриманим експериментально числовим інтервалам для відповідного сеансу реабілітації, отримують математичну задачу для обчислення оцінки вектора параметрів моделі у вигляді інтервальної системи нелінійних алгебричних рівнянь (ІСНАР) [38, 104]. Як правило, структура та параметри моделі є невідомими, тому необхідно розв'язувати дві задачі: структурної ідентифікації (визначення набору базисних функцій) та параметричної ідентифікації (оцінка числових значень параметрів для фіксованої структури).

Параметрична ідентифікація спрямована на знаходження числових значень вектора параметрів для вже обраної структури моделі. Традиційна постановка полягає у відшуканні такого набору параметрів  $\Theta$ , який мінімізує різницю між виходом моделі та експериментальними спостереженнями, що безпосередньо приводить до методів найменших квадратів та їх варіацій [92]. Для лінійних за

параметрами моделей ці методи мають закриту або квазізакриту форму та забезпечують швидку збіжність. Для слабо нелінійних систем застосовують градієнтні та квазідругі методи, що ґрунтуються на локальній апроксимації цільової функції та матриць чутливості моделі до параметрів. Однак для інтервальних дискретних моделей виникають дві фундаментальні проблеми: висока розмірність простору параметрів (кількість шуканих параметрів може сягати десятків та сотень) та інтервальна невизначеність даних і параметрів. Класичний МНК припускає, що похибка є випадковою і добре описується статистично, тоді як в інтервальних моделях маємо гарантовано обмежену невизначеність. Методи інтервального аналізу похибок розглядають задачу оцінювання параметрів як пошук множини  $\Theta$ , для якої модель може відтворити всі експериментальні дані в межах допустимих похибок вимірювання. Така постановка є гарантованою, але вона призводить до складних, часто негладких областей допустимих розв'язків у просторі параметрів.

Структурна ідентифікація розглядається як вибір класу та форми моделі системи, тобто визначення того, які змінні мають бути включені до моделі, яким є характер залежностей між ними, який порядок та тип динаміки слід враховувати. Для дискретних моделей із розподіленими параметрами структурна ідентифікація має особливу складність: необхідно не лише вирішити, які змінні та зв'язки є релевантними, але й визначити просторову структуру моделі. Найпростішим підходом до структурної ідентифікації є повний або частковий перебір можливих структур із подальшим порівнянням їхньої якості за інформаційними критеріями, наприклад критерієм Акаїке або байєсівським інформаційним критерієм. Проте для просторово-дискретизованих систем кількість потенційних зв'язків між компонентами стану зростає комбінаційно, що робить повний перебір практично непридатним. Другий класичний напрям – покроковий відбір регресорів (stepwise selection), але покрокові методи часто зупиняються на локально раціональній структурі, яка не є глобально оптимальною. Задачі структурної та параметричної ідентифікації на основі інтервальних даних відзначаються NP-обчислювальною складністю [38, 104], що робить класичні градієнтні методи оптимізації

непридатними через відсутність аналітичного вигляду функції мети та наявність множинних локальних мінімумів.

Альтернативою класичним методам є евристичні та метаевристичні алгоритми глобальної оптимізації, які розглядають задачу ідентифікації як пошук мінімуму цільової функції без вимог до її гладкості, опуклості чи диференційованості [98]. Ключовими представниками цього класу є генетичні алгоритми, алгоритм рою частинок та алгоритм бджолоїної колонії. Усі перелічені метаевристики мають спільну властивість: вони здійснюють глобальний пошук без припущень про лінійність або гладкість моделі. На відміну від класичних методів (МНК, градієнтних підходів), які в основному є локальними оптимізаторами і сильно залежать від початкового наближення, ройові та еволюційні алгоритми можуть віднайти множину прийнятних розв'язків, що відповідає природі інтервальної ідентифікації. Це критично важливо для дискретних моделей з розподіленими параметрами, оскільки метою часто є не стільки одна найкраща векторна оцінка, скільки множина узгоджених з даними параметрів, яка характеризує допустимі фізично правдоподібні стани системи.

Генетичні алгоритми (ГА) моделюють еволюційний процес добору, схрещування і мутації популяції кандидатних рішень [99]. Кожен можливий вектор параметрів  $\Theta$  інтерпретується як «особина», якій призначається пристосованість (fitness), що відображає якість збігу моделі з даними. Генетичні алгоритми добре працюють з дискретними й комбінаторними структурами, а також з обмеженнями типу «наявний/відсутній зв'язок», тому активно застосовуються для структурної ідентифікації (відбір релевантних регресорів, топологій зв'язків). У контексті параметричної ідентифікації вони дозволяють вільно досліджувати великий простір параметрів без вимоги до гладкості функції якості. Проте ГА часто вимагають ретельного налаштування операторів схрещування та мутації й можуть бути відносно «важкими» з точки зору збереження різноманіття популяції при великій кількості параметрів.

Алгоритм рою частинок (PSO – Particle Swarm Optimization) базується на колективній поведінці частинок, які «рухаються» простором параметрів,

орієнтуючись як на власний досвід пошуку кращих значень, так і на досвід сусідів або всієї популяції [100]. У контексті ідентифікації це означає, що багато кандидатних векторів параметрів паралельно перевіряються щодо якості відтворення експериментальних даних, і кожен кандидат адаптивно зміщує свої значення в напрямку більш успішних сусідів. Алгоритм рою частинок не потребує диференційовності цільової функції і добре масштабується у просторах великої розмірності, оскільки оновлення положення частинки є відносно простим обчислювально. Переваги алгоритму: простота реалізації, невисокі вимоги до кількості гіперпараметрів, добра поведінка у помірно великій розмірності. Недолік – схильність до передчасної конвергенції в локальний мінімум, якщо не вжити механізмів підтримання різноманіття [98].

Мурашині алгоритми базуються на спостереженні за тим, як колонії мурах знаходять шляхи до джерел їжі за допомогою феромонних слідів [101]. У моделі оптимізації кожен маршрут кодується як послідовність виборів, які мураха робить на графі; після побудови рішень мурахи підсилюють феромон на кращих шляхах. Мурашині алгоритми особливо добре працюють для дискретних комбінаторних задач (шляхи, топології, відбір структурних зв'язків) і масштабно використовуються у задачах структурної ідентифікації – зокрема для вибору топології моделі, лагів або кандидатних взаємодій між змінними. Однак класичний мурашиний алгоритм менш природно пристосований до безперервних параметрів, і його типова сильна сторона – саме комбінаторна частина, а не безпосередньо точна числова параметризація моделей.

Алгоритм штучної бджолиної колонії (ABC – Artificial Bee Colony), вперше запропонований Karaboga [102] та розвинений у подальших працях [105, 39], є метаевристичним методом глобальної оптимізації, що моделює поведінку медоносних бджіл при пошуку джерел нектару. Популяція агентів розділяється на ролі (робочі бджоли, розвідниці, спостерігачі), які відповідають за різні аспекти пошуку. Робочі бджоли вдосконалюють уже знайдені перспективні розв'язки, розвідниці досліджують нові області простору параметрів, а спостерігачі приймають рішення про те, які області варто посилено досліджувати далі, на основі

«якісності нектара», тобто значення функції мети. Цей поділ обов'язків забезпечує природний механізм балансу між уточненням та різноманіттям без явної потреби в складній параметризації динаміки руху [105]. АБК добре адаптується до обмежень, що кожен параметр має бути в інтервалі, тому природно підходить для інтервальних моделей.

У контексті задачі ідентифікації інтервальних моделей динаміки відновлення кутів рухливості суглобів алгоритм АВС застосовується для розв'язання двох взаємопов'язаних задач [104, 40]. Структурна ідентифікація полягає у пошуку оптимального набору базисних функцій серед множини моделей-претендентів: на кожній ітерації робочі бджоли модифікують поточну структуру заміною частини базисних функцій, а бджоли-дослідники обирають перспективні структури з імовірністю, пропорційною їхній якості. Параметрична ідентифікація для кожної зафіксованої структури забезпечує пошук оцінок вектора параметрів, що мінімізують функцію мети. Такий підхід забезпечує одночасну оптимізацію структури та параметрів моделі [104, 40]. Порівняно з іншими метаевристичними, АБК має перевагу у тому, що кожна бджола може інтерпретуватися як окремий програмний агент, який незалежно оцінює якість певної конфігурації параметрів або навіть певної структури моделі, що є важливим для подальшої агентно-орієнтованої програмної реалізації.

АВС успішно застосовувався для ідентифікації інтервальних моделей у задачах екологічного моніторингу шкідливих викидів автотранспорту [103, 46, 125], моделювання профілю користувача соціальних мереж [104] та інших прикладних задачах. У роботах Дивака та Кіндзерського [108] показано, що паралельна реалізація АВС на графічних процесорах (NVIDIA CUDA) дозволяє суттєво скоротити час обчислень для задач великої розмірності, а у [109] реалізовано метод структурної ідентифікації інтервальних моделей на основі поведінкової моделі бджолиної колонії з використанням попередньо розробленого CUDA-рішення для параметричної ідентифікації. Ці результати підтверджують, що АВС є масштабованим алгоритмом, який може ефективно працювати навіть для моделей з десятками параметрів за умови використання паралельних

обчислювальних архітектур. Для задачі ідентифікації моделей динаміки реабілітації, де кількість параметрів та структурних елементів є поміною, ABC забезпечує конвергенцію до глобального оптимуму за прийнятний час навіть на стандартних CPU, що є важливим для інтеграції у клінічну програмну систему.

Ключовою перевагою інтервального підходу порівняно з класичними стохастичними методами є те, що він не вимагає великих обсягів даних для оцінки статистичних характеристик розподілів похибок. Достатньо знати лише межі похибки вимірювань, що є реалістичним для безмаркерних систем гоніометрії, де точність нейромережевого детектора характеризується максимальною кутовою похибкою. Інтервальні моделі забезпечують гарантовані межі прогнозу: якщо модель сумісна з експериментальними інтервальними даними, то прогнозовані інтервали містять усі можливі траєкторії відновлення з урахуванням заданої невизначеності [38, 103]. Методи структурної та параметричної математичних моделей, що ґрунтуються на аналізі інтервальних даних, розглянуті у роботах Вошніна А.П., Бакана Г.М., Куссуль Н.Н., Дивака М.П. [38, 103, 104]. Це дозволяє формувати обережні та обґрунтовані рекомендації для фізіотерапевта, що є критичним у клінічній практиці.

Порівняльний аналіз розглянутих підходів дозволяє обґрунтувати вибір інтервального підходу з ABC-оптимізацією для задачі прогнозування динаміки відновлення рухливості суглобів. Систематизовані результати порівняння наведено в таблиці 1.3. Регресійний аналіз є інтерпретованим, але не враховує часові залежності та невизначеність вимірювань, а при малих вибірках (10-20 точок) дає ненадійні оцінки. ARIMA враховує часовий контекст, але потребує довгих рядів і припускає лінійність. Нейронні мережі здатні моделювати нелінійні залежності, але є непрозорими та потребують великих навчальних вибірок. GPR забезпечує імовірнісні оцінки, але обчислювально затратний і дає широкі довірчі інтервали при малій кількості даних. Натомість інтервальний підхід природно працює з короткими вибірками та обмеженою невизначеністю, забезпечує гарантовані межі прогнозу та є інтерпретованим, оскільки структура моделі (набір базисних функцій) має фізичний зміст і може бути пояснена клініцисту [97]. Додатковою

перевагою обраного підходу є те, що інтервальні різницеві моделі можуть бути безпосередньо інтегровані у рекомендаційну систему для фізичного терапевта: прогнозовані інтервали значень кутів суглобів на наступні сеанси використовуються як вхідні дані для аналізу динаміки відновлення.

Таблиця 1.3.

## Порівняльний аналіз методів моделювання динаміки реабілітації

Метод	Потреба в обсязі даних	Врахування невизначеності	Інтерпретованість	Придатність для коротких вибірок (10-20)	Моделювання нелінійності
Лінійна регресія [35]	Мала	Ні	Висока	Обмежена (перенавчання)	Ні
ARIMA / ARMA [94]	Середня (≥50 точок)	Ні	Середня	Низька	Ні
Нейронні мережі (LSTM / GRU) [34, 35]	Велика (≥100 точок)	Ні (точкові прогнози)	Низька	Низька (перенавчання)	Так
Гаусівські процеси (GPR) [37]	Середня	Так (довірчі інтервали)	Середня	Середня	Так (через ядро)
<b>Інтервальні моделі + ABC (запропонований підхід)</b>	Мала (10-20 точок)	Так (інтервали параметрів)	Висока	Висока	Так (поліноміальні базиси)

Аналіз (табл. 1.3) підтверджує, що інтервальний підхід з ABC-оптимізацією є єдиним, який одночасно працює на коротких клінічних вибірках (10-20 сеансів), враховує невизначеність вимірювань безмаркерної гоніометрії та забезпечує високу інтерпретованість результатів, що є критично важливим для систем підтримки клінічних рішень.

Системи підтримки клінічних рішень (CDSS – Clinical Decision Support Systems) у контексті реабілітації інтегрують прогностичні моделі з медичними знаннями для формування рекомендацій [41]. Виділяють два основних типи CDSS:

на основі правил (формалізовані клінічні протоколи) та на основі машинного навчання (нейромеревеві класифікатори, навчання з підкріпленням). У сучасній практиці важливим аспектом є пояснюваність (explainability) рекомендацій CDSS: клініцист повинен розуміти причини кожної рекомендації, щоб прийняти обґрунтоване рішення [42]. Це суттєво обмежує застосування «чорноящиківих» нейромеревевих CDSS у клінічній практиці та підсилює аргументацію на користь інтерпретованих підходів, таких як rule-based логіка у поєднанні з інтервальним прогнозуванням. Традиційний підхід до управління реабілітацією є реактивним: фізіотерапевт спостерігає за результатами минулих сеансів, аналізує дані та приймає рішення про зміни тільки після виявлення проблем. На противагу цьому, проактивний підхід використовує прогнозування для передбачення негативних трендів (плато, уповільнення прогресу, ризик перевантаження) та пропонує коригування до того, як вони призведуть до погіршення результатів [42].

Таким чином, для забезпечення ефективного прогнозування динаміки відновлення рухливості суглобів верхніх кінцівок в умовах коротких вибірок та обмеженої невизначеності вимірювань найбільш обґрунтованим є використання інтервальних різницевоїх моделей з ідентифікацією структури та параметрів на основі алгоритму бджолоїної колонії. Цей підхід забезпечує: роботу з короткими вибірками (10-20 сеансів); природне врахування невизначеності через інтервальне подання даних; гарантовані межі прогнозу; інтерпретованість структури моделі; можливість інтеграції з рекомендаційною системою для проактивного управління реабілітацією. Саме ці переваги визначили вибір інтервального підходу як математичної основи дисертаційного дослідження.

#### **1.4. Аналіз систем підтримки клінічних рішень та рекомендаційних систем у фізичній реабілітації**

Ефективність фізичної реабілітації верхніх кінцівок суттєво залежить від здатності терапевта своєчасно виявляти негативні тенденції у динаміці відновлення та адаптувати терапевтичний протокол до індивідуальних потреб конкретного пацієнта [2, 53]. Традиційний підхід до управління реабілітаційним процесом є переважно реактивним: рішення щодо зміни інтенсивності, складності або

характеру вправ приймаються на основі ретроспективного аналізу після завершення чергового сеансу або серії сеансів, що не дозволяє оперативно реагувати на уповільнення прогресу [62]. За даних умов виникає затримка між моментом, коли реабілітація починає бути неефективною, та моментом, коли терапевт це виявляє і коригує стратегію. Це призводить до непродуктивних сеансів, продовження загального курсу лікування, зниження мотивації пацієнта та неоптимального використання обмежених клінічних ресурсів. Проблема набуває особливої гостроти у контексті реабілітації верхніх кінцівок, де індивідуальна варіабельність відновлення є надзвичайно високою: пацієнти з однаковим діагнозом можуть демонструвати кардинально різну динаміку залежно від віку, супутніх патологій, ступеня ушкодження тканин та мотивації [53].

Альтернативою реактивному управлінню є проактивний підхід, заснований на використанні математичних моделей прогнозування динаміки відновлення для передбачення негативних трендів (плато, уповільнення прогресу, ризик перевантаження) та формування рекомендацій щодо корекції протоколу до того, як проблеми стануть клінічно значущими [94]. Для реалізації такого підходу необхідна інтеграція кількох компонентів: об'єктивне автоматичне вимірювання показників рухливості суглобів (підрозділ 1.1), математичне моделювання та прогнозування динаміки відновлення (підрозділ 1.3), система підтримки клінічних рішень для формування обґрунтованих рекомендацій та механізм реалізації корекції через адаптивне терапевтичне середовище (підрозділ 1.2). Перехід від реактивного до проактивного управління вимагає створення замкнутого контуру керування, де результати вимірювань автоматично подаються на вхід прогнозної моделі, а результати прогнозування є основою для формування рекомендацій. Lanotte та ін. [111] у систематичному огляді можливостей та викликів AI у реабілітаційній медицині підкреслюють, що найбільший потенціал AI полягає саме в інтеграції збору даних, прогнозування та підтримки прийняття рішень у замкнений контур, проте більшість існуючих систем реалізують лише окремі компоненти без наскрізної інтеграції.

Системи підтримки клінічних рішень (Clinical Decision Support Systems, CDSS) є програмними системами, що інтегрують медичні знання з даними конкретного пацієнта для формування персоналізованих рекомендацій щодо діагностики, лікування або моніторингу [41]. CDSS зарекомендували себе як ефективний інструмент підвищення якості медичної допомоги: систематичний огляд Sutton та ін. [41] продемонстрував, що впровадження CDSS зменшує кількість медичних помилок на 30-55% та покращує дотримання клінічних протоколів. Shortliffe та Sepulveda [42] підкреслюють, що ключовим фактором успіху CDSS у клінічній практиці є пояснюваність (explainability) рекомендацій: клініцист повинен розуміти причини кожної рекомендації, щоб прийняти обґрунтоване та відповідальне рішення. У контексті фізичної реабілітації CDSS можуть використовуватися для автоматизації прийняття рішень щодо зміни терапевтичної стратегії на основі об'єктивних даних, що особливо важливо в умовах дефіциту кваліфікованих фізіотерапевтів та зростаючого навантаження на реабілітаційні центри: за оцінками ВООЗ, щорічна глобальна потреба у реабілітаційних послугах зростає, а близько 2,4 мільярда людей мають стани здоров'я, що потребують реабілітації [2].

За архітектурним підходом CDSS поділяють на два основних типи: засновані на правилах (rule-based) та засновані на машинному навчанні (ML-based) [41, 42]. Rule-based CDSS кодують формалізовані клінічні протоколи у вигляді умовних правил «якщо-то» (if-then): наприклад, «якщо приріст кута згинання за останні три сеанси менший за 5°, рекомендувати збільшення інтенсивності вправ». Перевагою rule-based систем є повна прозорість та пояснюваність рішень, а також можливість верифікації правил клініцистами. Основним недоліком є негнучкість: фіксований набір правил не здатний адаптуватися до індивідуальних особливостей пацієнта та не використовує прогнозування для передбачення майбутньої динаміки [41]. ML-based CDSS використовують алгоритми машинного навчання (нейронні мережі, навчання з підкріпленням, ансамблеві методи) для автоматичного виявлення патернів у даних та формування рекомендацій. Проте ML-based системи мають суттєві обмеження: потреба у великих навчальних вибірках (сотні-тисячі

пацієнтів), низька інтерпретованість рішень («чорний ящик») та складність клінічної валідації [36]. Kierner та ін. [112] пропонують гібридні підходи, що поєднують переваги обох типів, проте у реабілітаційному контексті такі системи залишаються недостатньо дослідженими.

У ширшому контексті інформаційних систем рекомендаційні системи (Recommender Systems) – це клас програмних систем, що на основі аналізу даних пропонують користувачеві релевантні елементи або дії [114]. Ricci та ін. [114] класифікують рекомендаційні системи за методом формування рекомендацій: колаборативна фільтрація (на основі схожості між користувачами), контент-базована фільтрація (на основі характеристик елементів) та гібридні підходи. У медичному контексті рекомендаційні системи набувають специфічних особливостей: рекомендації повинні бути обґрунтованими та пояснюваними; система несе відповідальність за безпеку пацієнта; рішення мають враховувати клінічні протоколи та індивідуальні обмеження [41]. Mehrabadi та ін. [115] запропонували фреймворк PERFECT для персоналізованих рекомендацій фізичних вправ, що використовує алгоритм контекстуального бандита (contextual bandit) для адаптивного підбору інтенсивності навантаження на основі біометричних даних із фітнес-трекерів. Хоча PERFECT продемонстрував статистично значиме збільшення тривалості фізичних вправ ( $P < .001$ ), він орієнтований на здорових користувачів і не враховує специфіку клінічної реабілітації, де безпека та обмеження рухливості суглобів є критичними факторами.

Інтеграція прогностичних моделей з рекомендаційними системами у реабілітації залишається малодослідженою галуззю. Brunner та ін. [129] здійснили зовнішню валідацію динамічної прогностичної моделі для відновлення функцій верхньої кінцівки після інсульту на основі спільної моделі (joint model), що поєднує повздовжні вимірювання рухових функцій (Action Research Arm Test, ARAT) з логістичною кривою зростання. Модель продемонструвала хорошу дискримінаційну здатність ( $AUC = 0,81-0,89$  залежно від часової точки), проте має суттєві обмеження: прогнозування здійснюється на рівні бінарної класифікації (досягне/не досягне клінічно значущого порогу), а не кількісного передбачення

конкретних значень кутів рухливості; модель не інтегрована у рекомендаційну систему і не дає конкретних вказівок щодо корекції протоколу; не враховується інтервальна невизначеність вимірювань. Лі та ін. [36] у систематичному огляді моделей машинного навчання для прогнозування моторних функцій після інсульту зазначають, що більшість існуючих прогнозних моделей є ізольованими від клінічного процесу прийняття рішень і не інтегровані з системами підтримки для терапевтів. Проте сама ідея використання динамічних моделей, що оновлюються з кожним новим вимірюванням, є концептуально близькою до адаптивного циклу прогнозування, розглянутого у даній роботі.

Pelosi та ін. [113] запропонували використання навчання з підкріпленням (Reinforcement Learning, RL) для автоматичної адаптації параметрів VR-реабілітаційного середовища до індивідуальних потреб пацієнта. RL-агент, навчений на даних попередніх сеансів, змінював складність VR-завдань (швидкість, точність, амплітуду рухів) з метою підтримки пацієнта в зоні оптимального терапевтичного навантаження. Система продемонструвала здатність адаптувати складність вправ у реальному часі, проте має суттєві обмеження для клінічного впровадження: навчання RL-агента потребує великих обсягів даних (тисячі ігрових епізодів), що є нереалістичним для окремого пацієнта; рішення RL-агента є непрозорими, що суперечить вимогам пояснюваності у медичних CDSS [41, 42]; система адаптує лише складність типізованих VR-вправ і не інтегрує арт-терапевтичний компонент, що обмежує терапевтичний потенціал. Крім того, RL-агент оптимізує миттєву винагороду (виконання поточної вправи), а не довгострокову траєкторію відновлення, що є принципово іншим підходом порівняно з прогнозуванням на основі математичних моделей динаміки відновлення.

Систематичний огляд Abedi та ін. [117], опублікований у прj Digital Medicine, проаналізував застосування AI для віртуальної реабілітації у домашніх умовах. З 2172 унікальних записів лише 13 досліджень відповідали критеріям включення, що свідчить про ранній етап розвитку цієї галузі. Автори виявили, що більшість систем використовують AI переважно для розпізнавання рухів або класифікації вправ, тоді

як інтеграція AI для прогнозування результатів та формування адаптивних рекомендацій залишається рідкісною. Жодна з проаналізованих систем не поєднувала прогнозування динаміки відновлення з рекомендаційною системою для корекції терапевтичного протоколу. Це підтверджує наявність критичного розриву між можливостями AI та їх використанням у реабілітаційній практиці, зокрема у частині формування замкнутого контуру адаптивного управління реабілітацією.

Концепція замкнутого контуру реабілітації набуває все більшої уваги у науковій спільноті. Wang та ін. [118] запропонували інноваційну стратегію замкнутого контуру реабілітації дискінезії верхніх кінцівок після інсульту, що інтегрує роботизовану реабілітацію, сенсорику руху, нейронні мікрофлюїдні моделі та нейроелектроніку. Автори підкреслюють, що ефективна реабілітація вимагає безперервного зворотного зв'язку між вимірюванням стану пацієнта, прогнозуванням результатів та адаптацією терапії. Проте запропонований контур орієнтований на роботизовані реабілітаційні системи з апаратним контролем імпедансу, а не на VR/AR-терапію, і не інтегрує рекомендаційну систему для фізичного терапевта. Це підтверджує, що ідея замкнутого контуру є актуальною, проте її реалізація у контексті AR-арт-терапії з інтервальним прогнозуванням та генеративним AI є новою.

Окремим напрямком досліджень є застосування генеративних моделей штучного інтелекту для персоналізації арт-терапевтичного контенту. Shojaei та ін. [70] провели змішане дослідження серед практикуючих арт-терапевтів щодо використання AI-генерованих зображень у арт-терапії. Результати показали, що 67% опитаних арт-терапевтів вважають AI-генеровані зображення потенційно корисними як допоміжний інструмент для створення початкових шаблонів, проте підкреслюють необхідність клінічного контролю та адаптації до індивідуальних потреб пацієнта. Zubala та ін. [119] в інтегративному огляді позиціонування креативного AI у процесі арт-психотерапії виявили, що з понад 550 перевірених записів лише 10 досліджень відповідали критеріям включення, що підтверджує новизну цього напрямку. Автори зазначають, що текст-до-зображення технології генеративного AI (DALL-E, Stable Diffusion) уможливають унікальний тип ко-

творчості, де терапевт формулює вимоги до зображення текстовими описами, а AI-система генерує відповідний візуальний контент, що відкриває нові можливості для персоналізації терапевтичних завдань, зокрема у контексті реабілітації.

Принципово важливим для реалізації рекомендаційної системи в контексті AR-арт-терапії є встановлення зв'язку між просторовою структурою терапевтичного зображення та біомеханічним профілем рухів пацієнта. Iosa та ін. [67] досліджували «ефект Мікеланджело» – феномен покращення рухових функцій при виконанні мистецьких завдань у VR-середовищі, де естетична мотивація спонукає пацієнта до рухів більшої амплітуди та точності, ніж при виконанні стандартних терапевтичних вправ. Christiansen та ін. [66] показали, що тренування мистецьких навичок є здійсненням та потенційно ефективним підходом до сенсомоторної реабілітації після інсульту, де характер художнього завдання безпосередньо визначає тип та амплітуду рухів. Ці дослідження підтверджують гіпотезу, що просторова структура зображення – вертикальні елементи стимулюють згинання плечового суглоба з піднесенням руки, горизонтальні елементи стимулюють відведення та приведення, дрібні деталі стимулюють точні рухи зап'ястка – може бути використана як механізм непомітної для пацієнта корекції біомеханічного профілю рухів. Такий механізм є неінвазивним терапевтичним втручанням: пацієнт продовжує виконувати базове арт-терапевтичне завдання, не усвідомлюючи зміни терапевтичної стратегії, тоді як біомеханічний профіль його рухів суттєво трансформується завдяки іншому просторовому розподілу елементів [46].

Генеративні моделі (DALL-E, Stable Diffusion та їхні розвитки) здатні створювати контурні зображення-розмальовки на основі текстових описів, що відкриває можливість автоматичної генерації терапевтичних завдань з урахуванням біомеханічних потреб конкретного пацієнта. Ключовою ідеєю є побудова багатоступеневого AI-конвеєра: клінічна рекомендація (які рухи потрібно стимулювати) → велика мовна модель (LLM) трансформує рекомендацію у структурований текстовий опис зображення → генеративна модель створює контурне зображення-розмальовку для AR-полотна [70]. Такий підхід дозволяє

уникнути жорстко заданих шаблонів арт-завдань та забезпечити адаптацію візуального контенту до індивідуальних потреб реабілітації кожного пацієнта. При цьому генеративний AI виступає не як автономний агент, а як інструмент реалізації клінічної рекомендації, що зберігає контроль фізичного терапевта над терапевтичним процесом та забезпечує пояснюваність кожного етапу конвеєра.

Для визначення ступеня розриву між науковими досягненнями та клінічною практикою проведено порівняльний аналіз існуючих комплексних VR/AR-систем реабілітації верхніх кінцівок за шістьма критеріями (таблиця 1.4):

- тип вимірювання рухів;
- наявність прогностичних можливостей на основі математичних моделей;
- інтеграція VR/AR-терапії;
- наявність рекомендаційної системи або CDSS;
- компонент арт-терапії;
- замкнений адаптивний цикл керування реабілітацією.

Система Vidnova [43] реабілітації верхніх кінцівок, розроблена компанією WE/AR Studio у партнерстві з Cisco та Міністерством цифрової трансформації України. Система використовує гарнітуру Magic Leap та зовнішню камеру Azure Kinect для відстеження рухів суглобів. Пацієнти виконують завдання з малювання у віртуальних середовищах (гірські, морські пейзажі), а фізіотерапевти отримують біомеханічні дані та аналітику сеансів у реальному часі. Система також інтегрує психологічну оцінку емоційного стану пацієнта. Проходить пілотне тестування в одному з національних реабілітаційних центрів України. Вона поєднує гарнітуру AR Magic Leap 2 з автоматичним збором кутів рухів для арт-терапії, проте покладається на проприєтарні SDK та не має модулів прогнозування і рекомендаційної системи. Дослідження Tieri та ін. [44] підтвердило ефективність VR-арт-терапії ( $ES = 0.245$ ), але без автоматичної гоніометрії та прогностичної аналітики. RAPAEL Smart Glove (Neofect) використовує контактні сенсори без кількісної гоніометрії та прогнозування. Tugomotion VirtualRehab обмежений розпізнаванням жестів без вимірювання кутів суглобів. MindMotion GO (MindMaze) – авторизована FDA медична VR для реабілітації інсульту, але без арт-

терапії та прогностичного модуля. Pelosi та ін. [113] реалізували адаптивний RL-агент для VR-реабілітації, але без прозорості рішень та арт-терапевтичного компонента.

Таблиця 1.4.

Порівняльний аналіз існуючих VR/AR/MR-систем реабілітації верхніх кінцівок

Система / Дослідження	Безмаркерна багатокам. гоніометрія	Прогнозування (мат. моделі)	AR / VR / MR	Арт-терапія	Рекомендації для фізіотерапевта	Замкнутий адаптивний цикл
Brunner та ін. [129] (Joint model)	Ні (клінічне ручне)	Так (Joint model)	–	–	–	–
Pelosi та ін. [113] (RL-адаптація VR)	Вбудоване (VR)	–	VR	–	RL-адаптація складності	Частково
Mehrabadi та ін. [115]	IMU-сенсори	–	VR	–	–	–
Wang та ін. [118]	Роботизоване	ML	–	–	Так (замкн. контур)	Частково
Tieri та ін. [44] (VR-арт-терапія)	–	–	VR	Так (пасивна)	–	–
Abedi та ін. [117]	Різні (огляд)	AI-моделі	VR	–	Частково	–
Vidnova (WE/AR Studio) [43]	Depth-сенсор (Azure Kinect, 1 камера)	–	MR (Magic Leap)	Так (малювання у MR)	Аналітика сеансів (без прогнозу)	–
<b>Запропонована система</b>	Так (3 камери + адаптивна фільтрація)	Так (інтервальні моделі + ABC)	AR (Magic Leap 2)	Так (активна, генеративна)	Так (AI + генеративні моделі)	Так (повний цикл)

Порівняльний аналіз (табл. 1.4) наглядно демонструє, що жодна з існуючих систем не поєднує одночасно всіх шести ключових компонентів: безмаркерну багатокамерну гоніометрію, прогнозування на основі математичних моделей, доповнену реальність, арт-терапію, рекомендаційну систему для фізіотерапевта та замкнутий адаптивний цикл реабілітації. Найбільш наближеною за функціональністю є система Vidnova, яка поєднує MR-гарнітуру Magic Leap з depth-сенсором Azure Kinect та арт-терапевтичним контентом (малювання у

віртуальних середовищах). Однак Vidnova використовує лише один depth-сенсор замість багатокамерної безмаркерної системи, не включає математичних моделей прогнозування динаміки відновлення та не реалізує замкнутого адаптивного циклу з автоматичною генерацією терапевтичних зображень на основі прогнозу.

Таким чином, актуальною залишається задача інтеграції в єдину систему наступних компонентів, відсутніх у існуючих рішеннях. По-перше, безмаркерна багатокамерна гоніометрія на основі аналізу трьох відеопотоків з адаптивною фільтрацією (підрозділ 1.1): існуючі системи або використовують рішення із однією камерою, що мають обмежену точність, або покладаються на контактні датчики. По-друге, прогнозування динаміки відновлення на основі інтервальних математичних моделей з ідентифікацією алгоритмом бджолоїної колонії (підрозділ 1.3): жодна з проаналізованих систем не використовує інтервальне прогнозування для коротких клінічних вибірок. По-третє, AR-арт-терапія як терапевтична модальність (підрозділ 1.2): арт-терапевтичний компонент відсутній у більшості VR/AR-реабілітаційних платформ. По-четверте, рекомендаційна система для фізіотерапевта на основі прогнозованої динаміки: існуючі CDSS у реабілітації або працюють на основі фіксованих правил без прогнозування, або використовують непрозорі ML-методи. По-п'яте, механізм реалізації рекомендацій через автоматичну генерацію персоналізованого арт-терапевтичного контенту за допомогою генеративного AI: жодна система не інтегрує AI-генерацію зображень з реабілітаційними рекомендаціями. Інтеграція зазначених п'яти компонентів дозволяє побудувати замкнений адаптивний цикл управління реабілітаційним процесом (підрозділ 2.3).

Важливим аспектом побудови рекомендаційної системи для медичного застосування є забезпечення безпеки пацієнта та дотримання клінічних протоколів [34]. Рекомендації щодо збільшення інтенсивності рухової активності мають підпорядковуватися системі обмежень, що враховує максимально допустиму амплітуду для кожного типу руху залежно від діагнозу, стадії реабілітації та індивідуальних обмежень, встановлених фізичним терапевтом. При надмірно широкому інтервалі прогнозу, що свідчить про високу невизначеність, система

повинна рекомендувати консервативну стратегію зі збереженням поточних параметрів протоколу [116, 120]. Забезпечення пояснюваності рекомендацій є критично важливою вимогою: кожна рекомендація повинна супроводжуватися кількісним обґрунтуванням – прогнозованим значенням кута, очікуваною кількістю сеансів до досягнення цілі, відхиленням від стандартного курсу – що дозволяє фізичному терапевту критично оцінити доцільність запропонованої корекції [41, 42, 121]. Система має бути рекомендаційною, а не автономною: повна автоматизація прийняття терапевтичних рішень без участі клініциста є неприйнятною з етичного та регуляторного погляду [42].

На відміну від існуючих підходів, де адаптація здійснюється або вручну терапевтом без прогнозування, або автоматично, але непрозоро (RL-агент [113]), запропонований підхід формує пояснювані рекомендації з кількісним обґрунтуванням, що базуються на інтервальних прогнозних моделях (підрозділ 1.3), залишаючи остаточне рішення за фізичним терапевтом [42].

Таким чином, аналіз існуючих систем підтримки клінічних рішень та рекомендаційних систем у фізичній реабілітації виявив суттєвий розрив між потенціалом AI-технологій та їх інтеграцією у клінічну практику. Rule-based CDSS є прозорими, але негнучкими та не використовують прогнозування; ML-based системи здатні до адаптації, але непрозорі та потребують великих вибірок; існуючі VR/AR-реабілітаційні платформи реалізують лише окремі компоненти замкнутого контуру без наскрізної інтеграції вимірювання, прогнозування та рекомендацій. Інтеграція безмаркерної гоніометрії, інтервального прогнозування, AR-арт-терапії, рекомендаційної системи з пояснюваними рішеннями та генеративного AI для персоналізації терапевтичного контенту у замкнений адаптивний цикл залишається нерозв'язаною задачею, що визначає напрямок даного дисертаційного дослідження.

### **1.5. Постановка задачі дослідження**

Проведений у розділі 1 аналіз існуючих методів і програмних засобів для реабілітації верхніх кінцівок дозволив виявити ключові проблеми та обмеження сучасних підходів. У підрозділі 1.1 встановлено, що традиційні контактні методи вимірювання кутів суглобів (механічні гоніометри, інклінометри на смартфонах)

мають недостатню точність (5-10°) і не забезпечують автоматичного контролю, а маркерні системи motion capture, незважаючи на високу точність (1 мм), є непрактичними для клінічного використання через надзвичайну вартість. Безмаркерні системи на базі глибокого навчання (MediaPipe, OpenPose) показують перспективність, проте потребують адаптивного фільтрування шуму та багатокамерної тріангуляції для досягнення клінічно прийнятної точності.

У підрозділі 1.2 обґрунтовано, що технології VR/AR суттєво підвищують мотивацію пацієнтів та ефективність реабілітації, а поєднання VR з арт-терапією активує додаткові мотиваційні центри мозку та покращує як фізичні, так і психологічні результати. Однак інтеграція точних вимірювань кутів рухливості суглобів з VR/AR-терапією залишається спорадичною у літературі. У підрозділі 1.3 показано, що для прогнозування динаміки відновлення в умовах коротких вибірок (10-20 сеансів) та обмеженої невизначеності вимірювань найбільш обґрунтованим є використання інтервальних різницевих моделей з ідентифікацією на основі алгоритму бджолиної колонії, який забезпечує глобальний пошук без вимог до гладкості цільової функції. У підрозділі 1.4 виявлено, що існуючі системи підтримки клінічних рішень у реабілітації або працюють на основі фіксованих правил без прогнозування, або використовують непрозорі ML-методи, а жодна з проаналізованих VR/AR-реабілітаційних систем не поєднує водночас безмаркерну гоніометрію, інтервальне прогнозування, арт-терапію, рекомендаційну систему та генеративний AI у замкнений адаптивний цикл.

Виходячи із проведеного аналізу літературних джерел, сформульовано науково-технічне завдання, яке полягає у розробці комплексного програмного забезпечення системи AR-реабілітації верхніх кінцівок, що інтегрує безмаркерну гоніометрію на основі аналізу трьох відеопотоків з адаптивною фільтрацією, інтервальне математичне моделювання динаміки відновлення рухливості суглобів за допомогою алгоритму бджолиної колонії, AR-арт-терапію у середовищі доповненої реальності та рекомендаційну систему підтримки для фізіотерапевта з механізмом реалізації рекомендацій через генерацію персоналізованого арт-терапевтичного контенту.

Таким чином, метою дисертаційної роботи є підвищення ефективності реабілітації верхніх кінцівок шляхом розробки комплексного програмного забезпечення, що реалізує замкнений адаптивний цикл «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів» на основі інтеграції безмаркерної гоніометрії на основі аналізу трьох відеопотоків, інтервального математичного моделювання, AR-арт-терапії та рекомендаційної системи з генеративним AI.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. Провести аналіз існуючих методів і програмного забезпечення для вимірювання кутів суглобів, технологій VR/AR у реабілітації, методів математичного моделювання та прогнозування динаміки відновлення, а також систем підтримки клінічних рішень та рекомендаційних систем у фізичній реабілітації.

2. Розробити метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією координат ключових анатомічних точок та алгоритмом геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху.

3. Побудувати інтервальні математичні моделі динаміки відновлення кутів рухливості плечового суглоба на основі адаптованих методів структурної та параметричної ідентифікації з використанням алгоритму бджолиної колонії.

4. Спроекувати комплексну архітектуру медичної реабілітаційної програмної системи на основі оркестрації мікросервісів: аналізу трьох відеопотоків з пріоритизацією, прогнозування, рекомендацій на основі генеративного штучного інтелекту та застосування арт-терапії для гарнітури Magic Leap 2, а також вебзастосування фізіотерапевта.

5. Інтегрувати розроблені компоненти – автоматичну безмаркерну гоніометрію, інтервальне прогнозування та систему рекомендацій на основі генеративного штучного інтелекту – в єдиний замкнений адаптивний цикл прогнозування та корекції терапевтичного протоколу.

6. Реалізувати програмну систему, провести побудову інтервальних математичних моделей на клінічних даних реабілітаційних пацієнтів та апробувати адаптивний цикл прогнозування і корекції терапевтичного протоколу.

### **Висновки до розділу 1**

1. Проведено аналіз сучасних методів і програмних засобів вимірювання кутів суглобів верхніх кінцівок. Розглянуто контактні методи (механічні гоніометри, інклінометри на смартфонах, інерціальні датчики), маркерні системи motion capture та безмаркерні системи на базі глибокого навчання (OpenPose, BlazePose, MediaPipe, YOLOv8-pose). Встановлено, що контактні методи мають недостатню точність (5-10°), маркерні системи є непрактичними через надзвичайну вартість, а безмаркерні системи потребують адаптивного фільтрування шуму та багатокамерної тріангуляції для досягнення клінічно прийнятної точності. Обґрунтовано доцільність застосування адаптивної фільтрації та геометричного злиття вимірювань від трьох камер.

2. Проаналізовано технології VR/AR у реабілітації та їх поєднання з арт-терапією. Показано, що технології доповненої та віртуальної реальності суттєво підвищують мотивацію пацієнтів та ефективність реабілітації, а поєднання VR/AR з арт-терапією активує додаткові мотиваційні центри мозку та покращує як фізичні, так і психологічні результати. Виявлено, що «ефект Мікеланджело» та просторова структура арт-терапевтичних завдань безпосередньо визначають біомеханічний профіль рухів пацієнта. Однак інтеграція точних вимірювань кутів рухливості суглобів з VR/AR-терапією залишається спорадичною у літературі.

3. Проведено порівняльний аналіз методів математичного моделювання та прогнозування динаміки відновлення рухливості суглобів. Показано, що класичні методи (регресійний аналіз, ARIMA, нейронні мережі, гаусівські процеси) мають суттєві обмеження при застосуванні до коротких клінічних вибірок (10-20 сеансів). Обґрунтовано вибір інтервального підходу з ідентифікацією структури та параметрів моделей на основі алгоритму бджолиної колонії як найбільш адекватного для задач з обмеженою невизначеністю та малою кількістю спостережень.

4. Проаналізовано системи підтримки клінічних рішень (CDSS) та рекомендаційні системи у фізичній реабілітації. Виявлено, що rule-based CDSS є прозорими, але негнучкими та не використовують прогнозування, а ML-based системи здатні до адаптації, але непрозорі та потребують великих вибірок. Порівняльний аналіз існуючих VR/AR-реабілітаційних платформ (Vidnova, RARAE, Tyromotion, MindMotion GO) виявив критичний розрив: жодна система не поєднує водночас безмаркерну гоніометрію, інтервальне прогнозування, AR-арт-терапію, рекомендаційну систему та генеративний AI у замкнений адаптивний цикл управління реабілітацією.

5. Виходячи із проведеного аналізу, сформульовано науково-технічне завдання, яке полягає у розробці комплексного програмного забезпечення системи AR-реабілітації верхніх кінцівок з замкнутим адаптивним циклом «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів». Також сформульовано завдання дисертаційного дослідження.

## РОЗДІЛ 2

### МЕТОДИ БЕЗМАРКЕРНОЇ ГОНІОМЕТРІЇ ТА МАТЕМАТИЧНІ МОДЕЛІ ДИНАМІКИ ВІДНОВЛЕННЯ КУТІВ РУХЛИВОСТІ СУГЛОБІВ

У попередньому розділі проведено аналіз існуючих підходів до автоматичного вимірювання кутів рухливості суглобів, застосування технологій віртуальної та доповненої реальності у фізичній реабілітації, а також методів прогнозування динаміки відновлення рухових функцій. Виявлено обмеження безмаркерних систем гоніометрії із однією камерою, зумовлені проекційною похибкою, та обґрунтовано доцільність застосування багатокamerної конфігурації з адаптивною фільтрацією для підвищення точності вимірювань. Встановлено, що інтервальний підхід до моделювання динаміки відновлення дозволяє коректно враховувати невизначеність вхідних даних, а метаевристичні алгоритми колективного інтелекту є ефективним інструментом розв'язання задач ідентифікації таких моделей.

У даному розділі описано математичне забезпечення комплексної системи автоматичного вимірювання та прогнозування динаміки відновлення кутів рухливості суглобів верхніх кінцівок. Розглянуто метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків із застосуванням адаптивного фільтра, що забезпечує одночасно субпіксельне згладжування шуму нейромережевої детекції та миттєву реакцію на швидкі рухи пацієнта. Описано алгоритм геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху суглобів, що зменшує проекційну похибку з  $30-50^\circ$  до  $3-5^\circ$ .

Далі, розглянуто метод ідентифікації інтервальних моделей динаміки відновлення кутів рухливості суглобів на основі алгоритму бджолиної колонії. Сформульовано задачі структурної та параметричної ідентифікації інтервальних дискретних динамічних моделей у вигляді різницевих рівнянь, описано функцію мети та фази метаевристичного алгоритму пошуку оптимальних параметрів моделей-претендентів.

Завершальною частиною розділу є опис адаптивного циклу прогнозування та корекції терапевтичного протоколу, який інтегрує результати вимірювань та прогнозування у замкнений контур керування реабілітаційним процесом. Розглянуто критерії аналізу прогнозованої динаміки, механізм формування рекомендацій щодо корекції терапевтичного зображення та параметрів AR-сесії, а також багатоступеневий AI-контур автоматичної генерації арт-терапевтичного контенту.

Результати цього розділу опубліковано автором у працях [21, 46, 47, 48].

### **2.1. Метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків**

Об'єктивне кількісне вимірювання діапазонів рухів у суглобах (ROM - Range of Motion) є фундаментальною складовою процесу реабілітації верхніх кінцівок, оскільки саме динаміка зміни ROM визначає ефективність терапевтичних втручань та дозволяє коригувати план лікування [3]. Традиційна мануальна гоніометрія з використанням механічного кутоміра залишається найпоширенішим методом оцінки ROM у клінічній практиці (рисунок 2.1), проте має суттєві обмеження: між-спостерігальна надійність (inter-rater reliability) при вимірюванні плечового суглоба коливається в межах ICC = 0,65-0,84, що відповідає стандартній похибці 8-14° [72]. Така варіабельність означає, що зміна показника ROM менше ніж на 15° між двома спостерігачами не може бути інтерпретована як реальна динаміка відновлення. Крім того, мануальна гоніометрія дозволяє фіксувати лише статичні крайні положення суглоба, не забезпечуючи безперервного вимірювання під час виконання руху. Маркерні оптичні системи захоплення руху (Vicon, OptiTrack, Qualisys) забезпечують субміліметрову точність визначення координат при частоті до 2000 Гц [73], однак їх клінічне впровадження обмежене високою вартістю (50 000-150 000 USD), необхідністю спеціально обладнаного приміщення та тривалою підготовкою пацієнта (встановлення маркерів займає 20-40 хв).

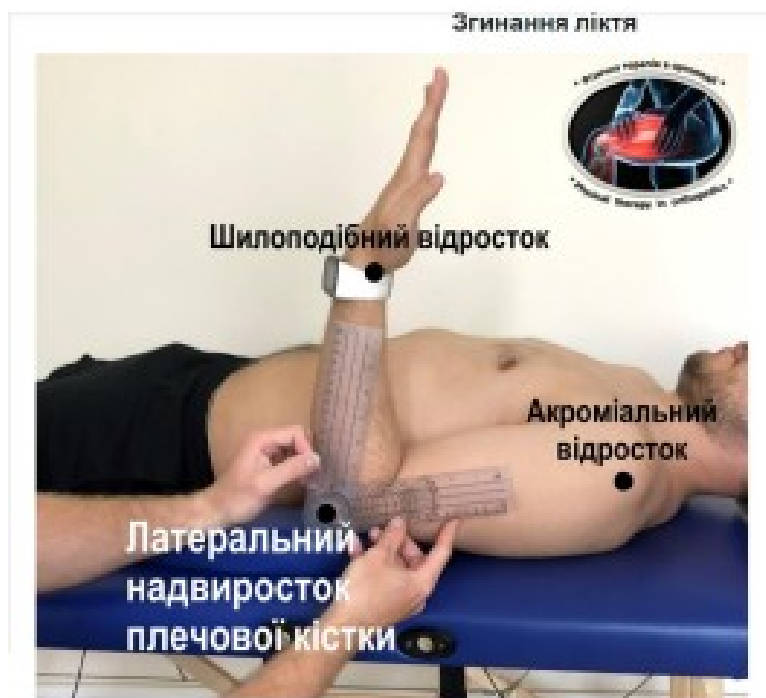


Рис. 2.1. Процес вимірювання кута згину суглоба з допомогою гоніометра.

Розвиток методів глибокого навчання для оцінювання пози людини (Human Pose Estimation, HPE) відкрив принципову можливість безмаркерного відеоаналізу на стандартних камерах [9, 12]. У сучасній літературі представлено низку HPE-моделей, що відрізняються за архітектурою, точністю та обчислювальними вимогами. OpenPose [12] реалізує підхід «знизу вгору» із використанням Part Affinity Fields для одночасної детекції кількох осіб, досягаючи MPJPE  $\approx$  65 мм на стандартних бенчмарках, проте вимагає GPU-прискорення для роботи в реальному часі. MoveNet [15] оптимізований для мобільних пристроїв із використанням нейронної архітектури MobileNetV2 та досягає балансу між швидкістю (30+ FPS на CPU) і точністю. ViTPose [16] на основі Vision Transformer демонструє найвищу точність серед сучасних моделей (MPJPE  $\approx$  37 мм), проте потребує значних обчислювальних ресурсів. Мого та ін. [123] у порівняльному дослідженні 2025 року оцінили 11 HPE-моделей для клінічного аналізу рухів і підтвердили, що MediaPipe забезпечує оптимальний баланс між точністю, швидкістю та доступністю для клінічного застосування без GPU. Бібліотека MediaPipe Pose Landmarker від Google [14, 131] реалізує детекцію 33 анатомічних точок тіла у реальному часі із використанням архітектури BlazePose [13]. Принципово важливою особливістю є

те, що модель повертає одночасно два представлення координат: нормалізовані 2D-координати у піксельному просторі зображення (pose2d) та тривимірні координати у метричному просторі відносно центру мас пацієнта (pose\_world\_3d). Перше представлення використовується для візуалізації скелету на відеокадрі та оцінки видимості точок у кадрі, тоді як друге - для геометрично коректного обчислення кутів суглобів, незалежного від відстані до камери. Додатково для підвищення точності вимірювань зап'ястних кутів використовується MediaPipe Hand Landmarker, що забезпечує детекцію 21 точки кисті з обох рук. Модель повертає показник видимості  $visibility \in [0, 1]$  для кожної точки, що характеризує впевненість детекції та використовується як вагова компонента у процедурі злиття вимірювань від трьох камер.

Порівняльний аналіз точності НРЕ-моделей показує, що MediaPipe досягає середньої похибки локалізації суглобів MPJPE  $\approx 52$  мм на стандартних 3D-бенчмарках [76], що транслюється в кутову похибку 3-8° залежно від відстані до камери та конфігурації суглоба. Ця точність є достатньою для клінічного моніторингу, де мінімальна клінічно значуща різниця (MCID) для більшості суглобів становить 5-15° [107]. Водночас стохастичність нейромережевої детекції (тремор координат між послідовними кадрами) потребує ефективної темпоральної фільтрації для отримання стабільних кутових вимірювань.

Ключовою проблемою безмаркерних систем із однією камерою є проекційна похибка - спотворення вимірюваного кута при невеликому ракурсі камери відносно площини руху суглоба. Для суглоба з реальним кутом  $\theta$ , виміряним камерою, оптична вісь якої відхилена від нормалі до площини руху на кут  $\varphi$ , спостережуваний кут  $\theta_{obs}$  визначається геометричним співвідношенням:

$$\theta_{obs} = \arccos(\cos(\theta) \cdot \cos^2(\varphi) + \sin^2(\varphi)) \quad (2.1)$$

При  $\varphi = 0^\circ$  проекційна похибка відсутня; при  $\varphi = 45^\circ$  і  $\theta = 90^\circ$  абсолютна похибка досягає 30°; при  $\varphi = 90^\circ$  вимірювання стає повністю неінформативним [88]. У попередніх роботах [21, 46] підтверджено принципову можливість автоматичного гоніометричного аналізу на базі нейронних мереж, однак

конфігурація із однією камерою залишала невирішеною проблему проекційної похибки для рухів поза фронтальною площиною.

Для усунення цього обмеження у даній роботі запропоновано метод безмаркерної гоніометрії, що базується на трьох IP-камерах, розташованих у взаємно перпендикулярних ракурсах: FRONT (фронтальна), LEFT та RIGHT (бічні сагітальні). Таке геометричне розміщення гарантує, що для кожного з основних анатомічних рухів верхніх кінцівок принаймні одна камера знаходиться поблизу площини руху з кутом відхилення  $\varphi < 20^\circ$ , що обмежує проекційну похибку значенням 3-5° [88]. Система вимірює 22 кутових параметри (11 на кожную сторону тіла), що охоплюють рухи плечового суглоба (згинання 160-180°, розгинання 50-60°, відведення 180°, приведення 30°, внутрішня та зовнішня ротація 70-90°/90°), ліктьового суглоба (згинання 140-160°, перерозгинання 8-12°, пронація 85-95°, супінація 85-90°) та зап'ястного суглоба (згинання 70-80°, розгинання 60-70°) відповідно до стандартів AAOS [124].

Алгоритм обробки відеопотоку, що реалізує описаний метод у поточній програмній системі, наведено на рисунку 2.2. На першому кроці здійснюється захоплення кадру з відповідного джерела. Система підтримує два режими: RTSP-режим для IP-камер, де застосовується стратегія burst drain - послідовне отримання кількох кадрів підряд з відкиданням буферизованих для мінімізації затримки між реальним рухом та моментом обробки; та USB-режим для підключених вебкамер, де кадр дзеркалюється горизонтально для компенсації природного відображення камери. Перед передачею кадру у нейромережевий детектор виконується підготовка для inference: масштабування до максимальної сторони 256-640 пікселів методом INTER\_AREA та конвертація кольорового простору BGR → RGB, що зменшує обчислювальне навантаження у 2-4 рази при вхідному відео 1280×720 без суттєвої втрати точності детекції.

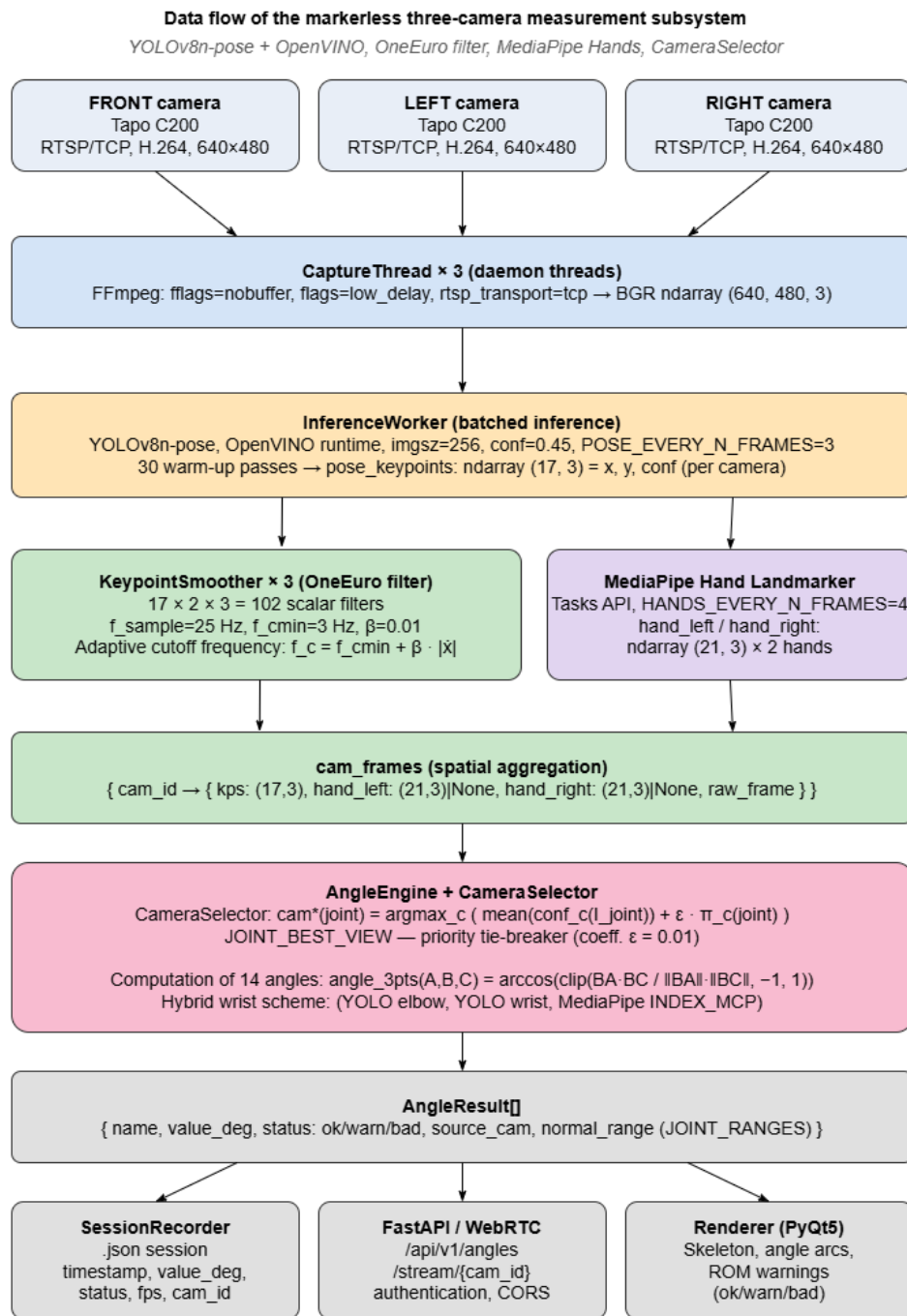


Рис. 2.2. Діаграма потоків даних мікросервісу автоматичної гоніометрії.

На другому кроці виконується неймережева детекція. MediaPipe Pose Landmarker запускається у режимі VIDEO (RunningMode.VIDEO), що дозволяє використовувати темпоральну когерентність між кадрами для стабілізації детекції. Результатом є одночасно pose2d – нормалізовані 2D-координати 33 точок тіла у площині зображення, та pose\_world\_3d – координати тих самих точок у тривимірному метричному просторі. Паралельно MediaPipe Hand Landmarker

виконує детекцію 21 точки кожної кисті. Оскільки Hand Landmarker повертає орієнтацію руки відносно зображення (а не відносно пацієнта), реалізовано алгоритм вирівнювання L/R-мітки: координати зап'ясть з детектора пози використовуються як опорні точки, і виконується оптимальне призначення за критерієм мінімальної суми евклідових відстаней між виявленими кистями та відомими позиціями зап'ясть.

Третім кроком є адаптивна фільтрація Калмана, що виконується незалежно для точок тіла та точок кистей. Для кожної анатомічної точки підтримується незалежний екземпляр фільтра KalmanLandmark зі вектором стану  $x_k = [p_x, p_y, v_x, v_y]^T$ , де  $p_x, p_y$  - координати точки у піксельному просторі,  $v_x, v_y$  - відповідні швидкості. Кінематична модель постійної швидкості описується матрицями переходу стану  $F \in \mathbb{R}^{4 \times 4}$  та спостереження  $H \in \mathbb{R}^{2 \times 4}$ :

$$F = [[1, 0, \Delta t, 0], [0, 1, 0, \Delta t], [0, 0, 1, 0], [0, 0, 0, 1]]$$

$$H = [[0, 0, 1, 0], [0, 0, 0, 1]]$$
(2.2)

де матриця  $F$  реалізує екстраполяцію позиції на основі поточної швидкості ( $p_k = p_{k-1} + v_{k-1} \cdot \Delta t$ ,  $v_k = v_{k-1}$ ), а матриця  $H$  проектує повний вектор стану на спостережувані координати. Стандартний цикл фільтра Калмана [19, 106] складається з етапу прогнозування:

$$\hat{x}_{k|k-1} = F \cdot \hat{x}_{k-1|k-1},$$

$$P_{k|k-1} = F \cdot P_{k-1|k-1} \cdot F^T + Q$$
(2.3)

та етапу оновлення:

$$S_k = H \cdot P_{k|k-1} \cdot H^T + R_k, \quad K_k = P_{k|k-1} \cdot H^T \cdot S_k^{-1},$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \cdot (z_k - H \cdot \hat{x}_{k|k-1}),$$

$$P_{k|k} = (I_4 - K_k \cdot H) \cdot P_{k|k-1}$$
(2.4)

де  $Q = q \cdot I_4$  - коваріація шуму процесу ( $q = 6 \cdot 10^{-4}$  для точок тіла,  $q = 8 \cdot 10^{-4}$  для точок кистей),  $P$  – коваріаційна матриця похибки оцінки стану,  $K_k$  - матриця

підсилення Калмана,  $S_k$  - коваріація нев'язки,  $z_k$  - спостережені піксельні координати.

Класичний фільтр Калмана з фіксованим параметром шуму вимірювань  $R$  створює нерозв'язний компроміс між згладжуванням та швидкістю реакції [122]: при малому  $R$  фільтр забезпечує швидку реакцію на зміни, але підсилює тремор у стаціонарному режимі; при великому  $R$  - ефективно згладжує шум, але вносить неприйнятну затримку при швидких рухах пацієнта. Для усунення цього компромісу запропоновано механізм адаптивного інноваційного вентиля, що динамічно змінює параметр шуму вимірювань  $r_k$  залежно від евклідової норми нев'язки  $d_k$  між поточним вимірюванням та прогнозом фільтра:

$$d_k = \| z_k - H \cdot \hat{x}_{\{k|k-1\}} \|_2 \quad (2.5)$$

Адаптивне правило зміни параметра шуму вимірювань визначається як:

$$\begin{aligned} r_k &= r_{base}, \text{ якщо } d_k \leq gate \\ r_k &= r_{base} \cdot (gate / d_k)^2, \text{ якщо } d_k > gate \end{aligned} \quad (2.6)$$

де  $gate = 35$  пікселів для точок тіла та  $gate = 25$  пікселів для точок кистей. Квадратична залежність забезпечує плавне зниження  $R$  при різких рухах: при  $d_k = 2 \cdot gate$  значення  $r_k$  зменшується у 4 рази порівняно з базовим, тобто вимірюванню довіряють у 4 рази більше; при  $d_k = 3 \cdot gate$  -  $r_k$  зменшується у 9 разів. Мінімальне обмеження  $r_k \geq 10^{-6}$  запобігає числовій нестабільності матричних обчислень. Фізична інтерпретація механізму наступна: різкий стрибок координат (велике  $d_k$ ) свідчить про реальний швидкий рух пацієнта, тому фільтр тимчасово збільшує довіру до нового вимірювання і миттєво відстежує нове положення кінцівки; після стабілізації руху ( $d_k < gate$ ) параметр  $R$  повертається до базового рівня  $r_{base}$ , забезпечуючи максимальне згладжування стохастичного шуму детектора. Додатковою практичною перевагою механізму є автоматична класифікація артефактів детекції: випадкові стрибки координат на 50-200 пікселів, спричинені тимчасовою плутаниною сторін тіла або самооклюзією, трактуються як великі нев'язки, що дозволяє фільтру швидко відновити правильну траєкторію без ручного втручання оператора.

На початку кожної сесії вимірювання виникає проблема «холодного старту»: вектор стану ще не ініціалізований, оцінки швидкостей відсутні, а коваріація  $P$  завищена. Для забезпечення стабільного перехідного процесу запропоновано процедуру адаптивного прогріву з лінійною інтерполяцією базового параметра шуму  $r\_base$  від початкового значення  $r\_high$  до стабільного  $r\_stable$  впродовж  $WARMUP\_FRAMES = 45$  кадрів ( $\approx 1,5$  с при 30 FPS):

$$\begin{aligned} r\_base(t) &= r\_high + (r\_stable - r\_high) \cdot t \\ t &= \min(1,0, \text{frame\_count} / 45) \end{aligned} \quad (2.7)$$

На початку прогріву ( $t = 0$ ) значення  $r\_base = r\_high$  забезпечує слабку фільтрацію, що дозволяє фільтру швидко адаптувати оцінки координат та швидкостей до реальних значень. По завершенні прогріву ( $t = 1$ ) параметр досягає стабільного значення  $r\_base = r\_stable$ , і фільтр переходить у режим агресивного згладжування. Обробка пропущених вимірювань реалізована через механізм прогнозування без оновлення: при оклюзії анатомічної точки ( $visibility < 0,30$  для точок тіла) виконується лише крок прогнозування (2.3) без кроку оновлення (2.4), коваріація  $P$  монотонно зростає, відображаючи зниження впевненості в оцінці. Після  $MAX\_MISSED$  послідовних кадрів без валідного вимірювання (3 кадри для точок тіла, 8 кадрів для точок кистей з огляду на їх вищу мобільність) фільтр повертає невизначений результат, що запобігає використанню ненадійних екстрапольованих координат у подальших обчисленнях кутів. Параметри адаптивного фільтра Калмана наведено у таблиці 2.1.

Таблиця 2.1.

Параметри адаптивного фільтра Калмана

Параметр	Pose	Hand	Інтерпретація
$r\_high$ (NOISE_HIGH)	0,35	0,30	Дисперсія шуму на початку (менша фільтрація)
$r\_stable$ (NOISE_STABLE)	0,06	0,05	Дисперсія шуму після прогріву (більша фільтрація)
$q$	$6 \times 10^{-4}$	$8 \times 10^{-4}$	Коваріація шуму процесу

Параметр	Pose	Hand	Інтерпретація
(PROCESS_NOISE)			Q
gate (GATE_PX)	35,0	25,0	Поріг інноваційного вентилля (пікселі)
MAX_MISSED	3	8	Максимум кадрів екстраполяції без вимірювань
WARMUP_FRAMES	45	45	Кількість кадрів фази прогріву ( $\approx 1,5$ с)

Четвертим кроком є побудова landmark map та обчислення кутів суглобів. На основі відфільтрованих координат будується єдиний словник points\_map, що об'єднує точки тіла (з префіксом P:), точки лівої кисті (LH:) та правої кисті (RH:). Для зап'ястних кутів додатково обчислюється допоміжна віртуальна точка HAND\_AXIS = (INDEX + PINKY) / 2, що апроксимує вісь долоні як середнє між координатами вказівного пальця та мізинця без необхідності використання окремого детектора. Далі система визначає, чи доступні тривимірні координати pose\_world\_3d для всіх трьох точок кутового трикутника конкретного суглоба. Якщо так - обчислення виконується у 3D метричному просторі, що повністю усуває проекційну похибку; якщо ні (наприклад, через недостатню видимість точки або вихід за межі кадру) - використовуються 2D піксельні координати з відповідним застереженням щодо можливої проекційної похибки. Для точок з *visibility* < 0,65 (поріг ANGLE\_MIN\_VISIBILITY) або нормалізованими 2D координатами поза межами кадру відповідний кут вважається недоступним і не включається у результат, що запобігає появі «примарних» кутів від невидимих або частково оклюдованих суглобів.

Обчислення кута суглоба виконується векторним методом для трьох анатомічних точок: A (проксимальна), B (вершина суглоба), C (дистальна). Формуються вектори  $BA = A - B$  та  $BC = C - B$ , після чого кут між ними обчислюється за формулою:

$$\theta = \arccos(\text{clip}((\mathbf{BA} \cdot \mathbf{BC}) / (\|\mathbf{BA}\| \cdot \|\mathbf{BC}\|), -1, 0, +1, 0)) \quad (2.8)$$

де функція  $\text{clip}()$  забезпечує числову стабільність при граничних значеннях аргументу  $\arccos$ , запобігаючи виникненню помилок через похибки числення з рухомою комою. Після обчислення геометричного кута  $\theta$  система визначає тип метрики для конкретного кутового параметра. Для параметрів з міткою *direct* результат  $\theta$  повертається безпосередньо як геометричний кут між ланками суглоба. Для параметрів з міткою *from\_straight*, що застосовується для вимірювання ліктьового суглоба та зап'ястка, застосовується додаткове перетворення:

$$\theta_{\text{metric}} = |180^\circ - \theta| \quad (2.9)$$

яке дає кут відхилення від повністю випрямленого положення кінцівки, що відповідає загальноприйнятій клінічній конвенції опису ROM. Наприклад, при повністю розігнутому лікті  $\theta = 180^\circ$  і  $\theta_{\text{metric}} = 0^\circ$ ; при згинанні до прямого кута  $\theta = 90^\circ$  і  $\theta_{\text{metric}} = 90^\circ$ . Результатом обчислення є об'єкт *AngleResult*, що містить значення кута у градусах, ознаку відповідності нормативному діапазону ROM згідно стандартів AAOS [124], та піксельні координати трьох точок для подальшої візуалізації дуги кута безпосередньо на відеокадрі.

П'ятим кроком є злиття вимірювань від трьох камер у режимі *multi-camera*. У режимі однієї камери результат *AngleResult* поточної камери використовується безпосередньо. На основі трьох відеопотоків для кожного з 22 кутових параметрів виконується зважений скоринг серед усіх доступних онлайн-камер:

$$\begin{aligned} \text{score} = & 0,50 \cdot \text{visibility} + 0,20 \cdot \text{geometry} + 0,15 \cdot \text{cam\_pref} + \\ & + 0,10 \cdot \text{readiness} + 0,05 \cdot \text{has\_person} + 0,08 \cdot \text{hysteresis} \end{aligned} \quad (2.10)$$

Компонента *visibility* є середньозваженою видимістю анатомічних точок кутового трикутника з урахуванням їх відстані від краю кадру - точки поблизу краю отримують знижений ваговий коефіцієнт (розраховується як мінімум нормалізованих відстаней до чотирьох країв кадру, нормований на поріг 0,08), оскільки детектор є менш надійним для частково обрізаних ділянок тіла. Ваговий коефіцієнт компоненти *visibility* розраховується як  $0,75 \cdot \text{mean}(\text{scores}) + 0,25 \cdot \text{min}(\text{scores})$ , що враховує як середню, так і найгіршу точку трикутника. Компонента

geometry оцінює геометричну якість вимірювання через довжину плечей кутового трикутника у пікселях та довжину хорди між крайніми точками - вироджені трикутники з короткими плечима (менше 45 пікселів) або малою хордою (менше 55 пікселів) отримують низьку оцінку, оскільки навіть незначна похибка позиціонування суглобової точки спричиняє велику кутову помилку. Компонента `cam_pref` реалізує апіорний пріоритет камери на основі геометрії площини руху суглоба: для рухів у сагітальній площині (згинання та розгинання) лівих суглобів пріоритет 1,0 отримує камера LEFT, потім FRONT з вагою 0,72 та RIGHT з вагою 0,35; для правих суглобів - симетрично RIGHT з вагою 1,0; для рухів у фронтальній площині (відведення, горизонтальне згинання, радіальне та ульнарне відхилення зап'ястка) пріоритет 1,0 отримує FRONT, потім бічна камера відповідної сторони з вагою 0,72 та протилежна з вагою 0,35. Компонента `readiness` враховує стан прогріву фільтра Калмана: камера у стані WARMING отримує знижений коефіцієнт  $0,35 + 0,65 \cdot \text{warmup\_progress}$ , що плавно зростає до 1,0 по завершенні прогріву, запобігаючи використанню нестабілізованих вимірювань на початку сесії. Компонента `has_person` приймає значення 1,0 якщо детектор виявив людину у кадрі і 0,0 в іншому випадку. Компонента `hysteresis = 0,08` є бонусом попередньо обраній камері, що забезпечує стабільність переключень між джерелами вимірювань і запобігає осциляції при рівних оцінках двох камер. Переможець обирається як `argmax(score)` по всіх доступних онлайн-камерах, а назва камери-переможця зберігається для наступної ітерації як попереднє джерело.

Шостим кроком є рендеринг результатів на відеокадрі: на зображення накладається скелет пацієнта з кольоровим кодуванням суглобів (зелений - кут у нормативному діапазоні ROM, синій - поза нормою), дуги кутів з числовими значеннями у градусах та індикатор прогресу прогріву фільтра. Для відображення україномовних підписів на OpenCV-кадрах використовується бібліотека Pillow з підтримкою TrueType-шрифтів з підтримкою кирилиці.

Сьомим, завершальним кроком є запис даних сесії. Якщо запис сесії активний, об'єкт `SessionRecorder` додає поточний кадр до потокового JSON-файлу із збереженням мітки часу, номера кадру, значення FPS та повного вектора з 22

кутових значень. Виклик `flush()` після кожного кадру гарантує збереження даних навіть при аварійному завершенні програми. По завершенні сесії зведення ROM-параметрів (мінімальне та максимальне спостережені значення для кожного з 22 кутів) передається через REST API серверній частині для збереження у базі даних PostgreSQL та подальшого інтервального моделювання динаміки відновлення (підрозділ 2.2). Якщо запис не активний - результат `AngleResult` повертається без збереження для використання виключно у режимі реального часу.

Описаний метод з адаптивним фільтром Калмана апробовано та опубліковано у попередній роботі автора [47]. На процесорі Intel Core i7-10700 (8 ядер) при обробці трьох потоків H.264 роздільної здатності 640×480 система забезпечує  $\approx 30$  FPS при завантаженні CPU 65-90%, середню абсолютну похибку (MAE)  $< 4^\circ$  та тремор  $< 1$  пікселя RMS. Водночас інтеграція методу у ширший контур AR-системи реабілітації - додавання AR-рендерингу для гарнітури Magic Leap 2, паралельних сесій декількох пацієнтів, серверної частини з модулями інтервального прогнозування та системи рекомендацій - виявила низку обмежень, пов'язаних із обчислювальною вартістю базової реалізації. По-перше, неймережевий інференс MediaPipe BlazePose full на CPU становить 12-18 мс/кадр і формує основне «вузьке місце» всього конвеєра; подальше зниження цього показника без переходу на GPU неможливе у межах TFLite-реалізації. По-друге, 99 незалежних екземплярів фільтра Калмана (33 точки  $\times$  3 камери), кожен з матричною операцією інверсії у етапі оновлення (2.4), створюють постійне фонове навантаження, що погано розпаралелюється між ядрами процесора через глобальне блокування інтерпретатора Python (GIL). По-третє, для типової клінічної робочої станції середнього класу без дискретної графіки (Intel Core i5 або AMD Ryzen 5) базова реалізація не забезпечує цільових 30 FPS на трьох камерах одночасно. По-четверте, зважений скоринг (2.10) з п'ятьма компонентами якості при близьких значеннях `confidence` для різних камер може призводити до перескоків між ракурсами, що проявляється у вигляді візуальних стрибків кутів значень на  $1-2^\circ$  між послідовними кадрами.

Для усунення цих обмежень у даній роботі розроблено оптимізований варіант реалізації методу, що зберігає принцип безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією координат ключових точок, однак використовує інший технологічний стек. Замість MediaPipe Pose Landmarker застосовано детектор YOLOv8n-pose від Ultralytics - нейромережеву модель на базі архітектури CSPDarknet з додатковою головою регресії ключових точок, що забезпечує виявлення людини та 17 анатомічних точок (COCO-keypoints) у єдиному прямому проході мережі. Варіант папо має розмір 6 МБ та обчислювальну вартість 9,2 GFLOPs, що у 3-5 разів менше за MediaPipe Pose full. Модель експортується у формат OpenVINO IR для прискореного виконання на CPU через векторизовані інструкції (AVX-512) та квантизацію FP16 з кешуванням компільованого графа, що скорочує час інференсу до 8-12 мс/кадр на тому ж процесорі i7-10700 - зменшення на 30-50%. Втрату деталізації кистей, яка у BlazePose була вбудованою, компенсує окремий виклик MediaPipe Hand Landmarker із субсемплінгом HANDS\_EVERY\_N\_FRAMES = 4 (кожен 4-й кадр), що є достатнім з огляду на повільну динаміку зап'ястних кутів у процесі реабілітації. Для детекції пози застосовується аналогічний субсемплінг POSE\_EVERY\_N\_FRAMES = 3 (кожен 3-й кадр), між якими значення інтерполюються згладжувальним фільтром. Такий підхід дозволяє знизити завантаження CPU при збереженні клінічно значущої точності вимірювань, оскільки кутові параметри верхніх кінцівок у процесі реабілітаційних вправ фізично не змінюються швидше  $\approx 5$  Гц.

Замість адаптивного фільтра Калмана з інноваційним вентилем для згладжування координат ключових точок у оптимізованій реалізації застосовано OneEuro-фільтр (Casiez, Roussel, Vogel, 2012) - скалярний низькочастотний фільтр з адаптивно-керованою частотою зрізу. Основна ідея фільтра полягає у динамічній зміні частоти зрізу  $f_c$  залежно від оцінки швидкості сигналу: при повільних рухах  $f_c$  мала (сильне згладжування, мінімізація тремору), при різких рухах -  $f_c$  зростає пропорційно модулю швидкості (слабке згладжування, швидкий відгук). Для

кожної координати ( $x$  або  $y$ ) кожної ключової точки підтримується незалежний екземпляр скалярного фільтра з рекурсивним експоненційним згладжуванням:

$$\hat{x}_k = \alpha_k \cdot x_k + (1 - \alpha_k) \cdot \hat{x}_{k-1}$$

$$\alpha_k = 1 / (1 + \tau_k / T_e), \quad \tau_k = 1 / (2\pi \cdot f_c),$$

(2.11)

$$T_e = 1 / f_{\text{sample}}$$

Оцінка швидкості зміни сигналу обчислюється як згладжена похідна з фіксованою частотою зрізу  $d_{\text{cutoff}} = 1,0$  Гц, після чого адаптивна частота зрізу  $f_c$  формується як лінійна функція цієї швидкості:

$$\hat{x}_k = \alpha_d \cdot (x_k - x_{k-1}) \cdot f_{\text{sample}} + (1 - \alpha_d) \cdot \hat{x}_{k-1},$$

$$f_c = f_{\text{cmin}} + \beta \cdot |\hat{x}_k| \quad (2.12)$$

де  $f_{\text{cmin}} = 3$  Гц - мінімальна частота зрізу при нульовій швидкості (забезпечує сильне згладжування у стаціонарному режимі),  $\beta = 0,01$  - коефіцієнт чутливості до швидкості (контролює, наскільки швидко фільтр «розслабляє» згладжування при зростанні швидкості),  $f_{\text{sample}} = 25$  Гц - цільова частота зчитування кадрів. При швидкому русі модуль швидкості  $|\hat{x}_k|$  зростає, відповідно зростає  $f_c \rightarrow \alpha_k \rightarrow 1$ , і фільтр практично без затримки пропускає вимірювання; у стаціонарному режимі  $|\hat{x}_k| \approx 0$ ,  $f_c = f_{\text{cmin}}$ ,  $\alpha_k$  мала, і фільтр агресивно згладжує тремор. Функціонально OneEuro-фільтр вирішує ту саму задачу компромісу між тремором та відгуком, що і адаптивний фільтр Калмана з інноваційним вентилем (2.5-2.7), однак досягає цього засобами скалярного експоненційного згладжування без вектора стану, без матричних операцій та без інверсії коваріаційної матриці. Обчислювальна складність OneEuro становить  $O(1)$  на одне вимірювання проти  $O(n^3)$  для Калмана при інверсії  $S_k$ , що у практичному вимірі забезпечує скорочення часу фільтрації з 0,5-1,0 мс (99 Калманів) до  $< 0,1$  мс (102 OneEuro-фільтрів) при еквівалентних клінічних характеристиках тремору та відгуку.

Алгоритм злиття вимірювань від трьох камер у оптимізованій реалізації також спрощено до детермінованої формули на базі аргмаксу:

$$\text{cam}^*(\text{joint}) = \underset{c \in \{ \text{FRONT}, \text{LEFT}, \text{RIGHT} \}}{\text{argmax}} \left( \langle \text{conf}_c \rangle \cdot \pi_c(\text{joint}) \right), \quad \varepsilon = 0,01 \quad (2.13)$$

де  $I_{\text{joint}}$  - множина індексів ключових точок, що беруть участь у обчисленні кута для суглоба  $\text{joint}$ ;  $\langle \text{conf}_c \rangle$  - середня впевненість детекції цих точок з камери  $c$ ;  $\pi_c(\text{joint}) \in \{0, 1, 2\}$  - ранг камери у пріоритетному списку  $\text{JOINT\_BEST\_VIEW}$  для даного суглоба, який використовується як тай-брейкер при близьких значеннях confidence (з малим коефіцієнтом  $\varepsilon = 0,01$ , що виключає його переважання над основним критерієм). Ключова відмінність від зваженого скорингу (2.10) полягає у тому, що після вибору камери усі три точки кутового трикутника беруться з однієї камери, що виключає помилки змішування проєкцій; одночасно виключається можливість осциляцій вибору між кадрами при близьких оцінках confidence, оскільки пріоритет камери є фіксованою константою для кожного суглоба. Такий підхід є простішим для клінічної інтерпретації - рецензенту достатньо показати, що система для кожного типу руху обирає камеру, з якою ці анатомічні точки найкраще видно.

Окрім аргмакс-вибору однієї камери, в оптимізованій реалізації додатково застосовано фільтр Калмана у ролі фузійного оцінювача (fusion estimator) для об'єднання вимірювань від трьох камер у єдину оптимальну оцінку координат кожної анатомічної точки. На відміну від базової реалізації, де фільтр Калмана виконує темпоральне згладжування окремих потоків (формули 2.3-2.7), тут він вирішує задачу просторової фузії, тобто оптимального зважування одночасних вимірювань однієї точки від декількох сенсорів з різною надійністю.

Для кожної ключової точки  $j$  ( $j = 0, \dots, 16$  для COCO-keypoints) підтримується незалежний екземпляр фузійного фільтра зі скороченим вектором стану  $x_j = [p_x, p_y]^T$ , що містить лише координати без компонент швидкості, оскільки темпоральна динаміка вже оброблена OneEuro-фільтром на попередньому етапі конвеєра. Модель процесу є статичною:

$$F_f = I_2, \quad Q_f = q_f * I_2 \quad (2.14)$$

де  $q_f = 10^{(-4)}$  - мінімальний шум процесу, що запобігає виродженню коваріаційної матриці  $P$  без внесення штучної динаміки між кадрами.

На кожному кадрі фузійний фільтр виконує етап прогнозування  $x_{\{k|k-1\}} = x_{\{k-1|k-1\}}$ ,  $P_{\{k|k-1\}} = P_{\{k-1|k-1\}} + Q_f$ , а потім послідовно оновлюється вимірюваннями від усіх камер, що мають валідну детекцію точки  $j$  (впевненість  $\gamma_{\{c,j\}} \geq \text{MIN\_KP\_CONF}$ ). Для камери  $c$  матриця шуму вимірювань визначається як обернено-квадратична функція впевненості детектора

$$R_{\{c,j\}} = (r_f / \gamma_{\{c,j\}}^2) * I_2 \quad (2.15)$$

де  $r_f = 0,5$  - базовий параметр шуму,  $\gamma_{\{c,j\}}$  - впевненість детектора YOLOv8-pose для точки  $j$  з камери  $c$ . Квадратична залежність  $1/\gamma_{\{c,j\}}^2$  забезпечує нелінійне зростання вагового вкладу камери зі зростанням впевненості: камера з  $\gamma = 0,9$  отримує  $R_{\{c,j\}} = 0,617$ , тоді як камера з  $\gamma = 0,4$  -  $R_{\{c,j\}} = 3,125$ , тобто її вплив на зливу оцінку у 5 разів менший.

Послідовне оновлення стану для кожної доступної камери  $c$  виконується за стандартними рівняннями фільтра Калмана:

$$\begin{aligned} S_c &= P + R_{\{c,j\}}, \quad K_c = P * S_c^{-1}, \quad x_j \leftarrow x_j + K_c * (z_{\{c,j\}} - x_j), \\ P &\leftarrow (I_2 - K_c) * P \end{aligned} \quad (2.16)$$

де  $z_{\{c,j\}} = [x_{\{c,j\}}, y_{\{c,j\}}]^T$  - згладжені OneEuro координати точки  $j$  з камери  $c$ . Оскільки  $P$  та  $R_{\{c,j\}}$  є діагональними матрицями, інверсія  $S_c$  виконується аналітично поелементно. Після послідовної обробки всіх доступних камер (від 1 до 3) вектор  $x_j$  містить оптимальну зважену оцінку координат, а діагональні елементи  $P$  відображають залишкову невизначеність зливої оцінки.

Ключовою властивістю фузійного підходу є те, що впевненість детектора  $\gamma_{\{c,j\}}$  імпліцитно кодує геометричну якість ракурсу камери для конкретної анатомічної точки: при русі у сагітальній площині фронтальна камера спостерігає рух перпендикулярно і видає високу впевненість, тоді як бічні камери бачать рух у проекції і видають знижену впевненість. Таким чином, фузійний фільтр автоматично зважує внесок кожної камери відповідно до її інформативності для кожного конкретного руху та кожної конкретної ключової точки на кожному кадрі.

Порівняно з аргмакс-вибором однієї камери, фузійний підхід має три суттєві переваги. По-перше, інформація від усіх камер використовується одночасно замість відкидання двох із трьох вимірювань; при трьох камерах з однаковою впевненістю дисперсія зливої оцінки зменшується у 3 рази порівняно з оцінкою за однією камерою. По-друге, відсутність дискретного переключення між камерами усуває стрибки кутових значень на 1-2 градуси між сусідніми кадрами, характерні для аргмакс-підходу при близьких оцінках впевненості двох камер. По-третє, при тимчасовій оклюзії точки на одній камері фільтр автоматично перерозподіляє ваги на решту камер без будь-якого зовнішнього механізму перемикавання.

Обчислювальна вартість фузійного Калмана є незначною: стан розмірності  $2 \times 1$  та коваріація  $2 \times 2$  потребують лише скалярних операцій для аналітичної інверсії діагональної матриці  $S_c$ , а загальна кількість екземплярів фільтра становить 17 (по одному на ключову точку СОСО). Сумарний час фузії для всіх 17 точок  $\times$  3 камери становить менше 0,05 мс на кадр. Параметри фузійного фільтра Калмана наведено у таблиці 2.2.

Таблиця 2.2.

Параметри фузійного фільтра Калмана

Параметр	Значення	Інтерпретація
$q_f$	$10^{-4}$	Шум процесу (запобігає виродженню $P$ )
$r_f$	0,5	Базовий параметр шуму вимірювань
К-сть екземплярів	17	По одному на ключову точку СОСО
Складність / точку	$O(1)$	Аналітична інверсія діагональної $2 \times 2$
Сумарний час	< 0,05 мс	На кадр, 17 точок $\times$ 3 камери

Узагальнене порівняння базової та оптимізованої реалізацій методу за основними технічними та обчислювальними характеристиками наведено у таблиці 2.3. Обидві реалізації зберігають ключові принципи методу: безмаркерної гоніометрії на основі аналізу трьох відеопотоків взятих у взаємно перпендикулярних ракурсах, адаптивне згладжування координат ключових точок, вибір оптимальної камери для кожного типу руху та обчислення кутів векторним

методом за формулою (2.8). Різниця полягає виключно у технологічному стеку реалізації цих принципів, що дозволяє застосовувати метод на різних апаратних платформах залежно від доступних обчислювальних ресурсів.

Таблиця 2.3.

## Порівняння базової та оптимізованої реалізацій методу

Компонент	Базова реалізація	Оптимізована реалізація
<b>НРЕ-детектор</b>	MediaPipe Pose Landmarker (full)	YOLOv8n-pose
Кількість точок тіла	33 (BlazePose)	17 (COCO) + MediaPipe Hands 21×2
Runtime	TFLite (CPU)	OpenVINO IR + кешування графа
Інференс, мс/кадр	12-18	8-12
Фільтр згладжування	Адаптивний Kalman ( $x, P, F, H, Q, R$ )	OneEuro ( $f_c = f_{cmin} + \beta \cdot  \dot{x} $ )
Обчислювальна складність фільтра	$O(n^3)$ - інверсія $S \in \mathbb{R}^{2 \times 2}$	$O(1)$ - скалярна рекурсія
Кількість екземплярів фільтра	99 ( $33 \times 3$ ) матричних	102 ( $17 \times 2 \times 3$ ) скалярних
Адаптивність	Innovation gate: $r_k \propto 1 / d_k^2$	Cutoff: $f_c = f_{cmin} + \beta \cdot  \dot{x} $
Злиття трьох камер	Зважений 5-компонентний скоринг (формула 2.10)	$\text{argmax}(conf) + \text{priority tie-breaker}$ (формула 2.13)
Частота інференсу пози	Кожен кадр (~30 FPS)	POSE_EVERY_N_FRAMES=3
Завантаження CPU (3 камери)	65-90% на i7-10700	35-55% на i7-10700 (оцінка)
Цільова платформа	Високопродуктивна станція з 8+ ядер	Стандартна клінічна станція без GPU
Наукова складова	Адаптивний innovation gate - авторський внесок	Інженерна адаптація методу до апаратних обмежень

Порівняльний аналіз запропонованого адаптивного фільтра Калмана з альтернативними методами фільтрації підтверджує його переваги. Ковзне середнє ( $N = 5$ ) забезпечує мінімальний тремор, але вносить затримку 2-3 кадри та знижує пікову амплітуду руху на  $8-12^\circ$  через усереднення екстремумів. Класичний фільтр Калмана з фіксованим  $R = 0,06$  демонструє мінімальний тремор у статиці, але затримує відгук на стрибок у 80 пікселів на 3-4 кадри; при  $R = 0,01$  відгук забезпечується за 1 кадр, однак тремор зростає до 3-5 пікселів RMS. Запропонований фільтр з інноваційним вентилем досягає тремору менше 1 пікселя RMS при одночасному відгуку за 1 кадр, що є недосяжним для класичних фільтрів з фіксованими параметрами. Оптимізований OneEuro-фільтр, попри суттєво простішу математичну модель, досягає клінічно еквівалентних показників тремору та відгуку за рахунок адаптивної частоти зрізу, що автоматично регулюється відповідно до поточної швидкості зміни сигналу.

Для кількісної верифікації точності системи проведено серію експериментів із жорсткими геометричними еталонами (кутові шаблони  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$ ,  $150^\circ$ ): середня абсолютна похибка (MAE) для оптимального ракурсу камери склала  $1,8^\circ$  ( $SD = 0,9^\circ$ ), при відхиленні ракурсу на  $45^\circ$  MAE зростає до  $18,3^\circ$ , а при  $70^\circ$  - до  $42,7^\circ$ . Конфігурація з трьома камерами із зваженим скорингом забезпечує середню абсолютну похибку менше  $4^\circ$  для всіх типів рухів верхніх кінцівок, що відповідає вимогам клінічного моніторингу реабілітації [107] та покращує точність у 10-17 разів порівняно з системою на базі однієї камери [21].

Таким чином, розроблений метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків із адаптивною фільтрацією координат забезпечує автоматичне, об'єктивне та безперервне вимірювання 22 кутових параметрів верхніх кінцівок у реальному часі. Метод усуває фундаментальне обмеження систем із однією камерою - проєкційну похибку - шляхом геометрично обґрунтованого злиття вимірювань від трьох ортогонально розташованих камер. Базова реалізація з адаптивним фільтром Калмана з інноваційним вентилем (2.5-2.7) та процедурою прогріву (2.7) вирішує компроміс між стабільністю та швидкістю реакції, характерний для класичних фільтрів із фіксованими параметрами, досягаючи

тремору менше 1 пікселя RMS при відгуку за 1 кадр. Оптимізована реалізація з OneEuro-фільтром (2.11-2.12) та детермінованим аргмакс-вибором камери (2.13) зберігає функціональні характеристики методу при суттєвому зниженні обчислювальних вимог, що робить систему придатною для розгортання на стандартному клінічному обладнанні без GPU-прискорення.

Отримані результати вимірювань ROM використовуються як вхідні дані для методу інтервального прогнозування динаміки відновлення рухливості суглобів, що описується у наступному підрозділі.

## 2.2. Інтервальні моделі динаміки відновлення кутів рухливості суглобів верхніх кінцівок

Розглянемо постановку задачі ідентифікації інтервальних математичних моделей динаміки відновлення кутів рухливості суглобів верхніх кінцівок.

Математичну модель динаміки відновлення кутів рухливості суглобів верхніх кінцівок розглядаємо, як дискретне (різницеве) рівняння у такому вигляді [125]:

$$v_{i,k}(\vec{V}) = f_1(\vec{V}) * g_1 + f_2(\vec{V}) * g_2 + f_m(\vec{V}) * g_m$$

$$i=1, \dots, I, k=d, \dots, K, \quad (2.17)$$

де  $v_{i,k}$  - означає модельоване значення кута  $i$ -го суглобу;  $\vec{g}$  - вектор параметрів моделі, значення яких необхідно оцінити на основі інтервальних даних;  $f_1(\vec{V}), f_2(\vec{V}), f_m(\vec{V})$  - набір базисних функцій;  $k$ - часова дискрета;  $d$  - порядок різницевого рівняння.

Враховуючи (2.17) рекурентне різницеве рівняння, для кожної базисної функції матимемо такий вектор:

$$\vec{V} = (v_{i,k-d}, v_{i,k-d+1}, \dots, v_{i,k})^T \quad (2.18)$$

Нехай у певний спосіб знайдено оцінки  $\hat{g}$  вектора параметрів  $g$ . Якщо задати початкові умови для різницевої схеми (2.17) у вигляді інтервалів  $[\hat{v}_{i,k}]$ ,  $i=1, \dots, I$ ,  $k=1, \dots, d-1$ , тоді математична модель динаміки відновлення кутів рухливості суглобів верхніх кінцівок матиме вигляд інтервального різницевого рівняння:

$$[v_{i,k}(\vec{\hat{V}})] = [f_1(\vec{\hat{V}})] * \hat{g}_1 + [f_2(\vec{\hat{V}})] * \hat{g}_2 + \dots + f_m(\vec{\hat{V}}) * \hat{g}_m$$

$$i=1, \dots, I, k=d, \dots, K, \quad (2.19)$$

де  $[\vec{\hat{V}}] = [\hat{v}_{i,k-d}, \dots, \hat{v}_{i,k}, \dots, \hat{v}_{i,k}]$  - інтервальний вектор з компонентами, які означають обчислені інтервальні оцінки  $[\hat{v}_{i,k}^-, \hat{v}_{i,k}^+]$  значень кутів рухливості суглобів.

В подальшому розгляді, не порушуючи його загальності, у математичних співвідношеннях усуваємо індекс  $i$  (що означає номер кута згину суглоба), оскільки для кожного окремого кута суглоба будуємо окрему інтервальну дискретну динамічну модель.

Спираючись на гіпотезу, що обчислені інтервальні оцінки  $[v_k^-, v_k^+]$  модельованої характеристики мають належати отриманим експериментально числовим інтервалам  $[z_k^-, z_k^+]$  цієї ж характеристики кута рухливості суглоба та для відповідного сеансу реабілітації  $k=1, \dots, K$ , отримуємо математичну задачу для обчислення оцінки  $\hat{g}$  вектора параметрів моделі [126]:

$$\begin{cases} [\hat{v}_0^-, \hat{v}_0^+] \subseteq [z_0^-, z_0^+], \dots, \\ [\hat{v}_{d-1}^-, \hat{v}_{d-1}^+] \subseteq [z_{d-1}^-, z_{d-1}^+] \\ z_k^- \leq [f_1(\vec{\hat{V}})] * \hat{g}_1 + [f_2(\vec{\hat{V}})] * \hat{g}_2 + \dots + f_m(\vec{\hat{V}}) * \hat{g}_m \leq z_k^+ \\ k=d, \dots, K \end{cases} \quad (2.20)$$

Отримана система є інтервальною системою нелінійних алгебричних рівнянь (ІСНАР) відносно невідомих компонент вектора  $\vec{g}$  оцінок параметрів моделі динаміки відновлення кутів рухливості суглобів верхніх кінцівок. Причому, у цій ІСНАР вирази  $[\hat{v}_0^-, \hat{v}_0^+] \subseteq [z_0^-, z_0^+], \dots, [\hat{v}_{d-1}^-, \hat{v}_{d-1}^+] \subseteq [z_{d-1}^-, z_{d-1}^+]$  - це початкові умови для різницевої схеми (2.19).

Математичний вираз (2.19) називаємо інтервальною дискретною моделлю динаміки відновлення кутів рухливості суглобів верхніх кінцівок.

Як правило, структура і параметри цієї моделі є невідомими. Тому потрібно розв'язувати дві задачі, а саме структурної та параметричної ідентифікації інтервальної дискретної моделі.

Розглянемо задачу структурної ідентифікації інтервальних моделей динаміки відновлення кутів рухливості суглобів верхніх кінцівок.

На відміну від задачі параметричної ідентифікації, вектор базисних функцій в задачі структурної ідентифікації є невідомим.

Для представлення математичної моделі динаміки уведемо поняття множини структурних елементів:

$$\lambda_s = \left\{ f_1^s(v_{k-d}, \dots, v_{k-1}) * g_1^s, \dots, f_m^s(v_{k-d}, \dots, v_{k-1}) * g_m^s \right\} \quad (2.21)$$

У виразі  $s$  - означає певний набір структурних елементів, на основі якого будуємо  $s$ -ту модель у вигляді (2.21), тоді  $\lambda_s$  використовуємо для позначення  $s$ -ої структури.

Математичні моделі, які розглядатимемо у процесі структурної ідентифікації мають такий вигляд (2.22):

$$v_k(\lambda_s) = f_1^s(v_{k-d}, \dots, v_{k-1}) * g_1^s + \dots + f_m^s(v_{k-d}, \dots, v_{k-1}) * g_m^s \quad (2.22)$$

Зважаючи на отримані експериментальні інтервальні дані, задаємо умову узгодженості математичної моделі із отриманими інтервальними даними:

$$v_k(\lambda_s, V_k) \in [z_k^-, z_k^+], \quad \forall k=0, \dots, K \quad (2.23)$$

де  $z_k^-, z_k^+$  - нижня і верхня межа експериментально отриманих значень характеристики,  $v_k(\lambda_s, V_k)$  - означає істинне значення вихідної характеристики для фіксованого набору структурних елементів  $\lambda_s$  і для фіксованих значень вектора  $\vec{V} = (v_{k-d}, \dots, v_{k-1})^T$ , у часових дискретах  $k=0, \dots, K$ .

Приймаючи до уваги умову (2.23) отримаємо наступну систему:

$$\begin{cases} z_0^- \leq f_1^s(\vec{V}_0) * g_1^s + \dots + f_m^s(\vec{V}_0) * g_m^s \leq z_0^+ \\ \vdots \\ z_{d-1}^- \leq f_1^s(\vec{V}_{d-1}) * g_1^s + \dots + f_m^s(\vec{V}_{d-1}) * g_m^s \leq z_{d-1}^+ \\ z_d^- \leq f_1^s(\vec{V}_d) * g_1^s + \dots + f_m^s(\vec{V}_d) * g_m^s \leq z_d^+ \\ \vdots \\ z_K^- \leq f_1^s(\vec{V}_K) * g_1^s + \dots + f_m^s(\vec{V}_K) * g_m^s \leq z_K^+ \end{cases} \quad (2.24)$$

Спираючись на той факт що в отриманій ІСНАР перші  $d$  інтервальних рівнянь це початкові умови, то перепишемо її у наступному вигляді

$$\begin{cases} [v_0^-, v_0^+] \subseteq [z_0^-, z_0^+], \dots, [v_{d-1}^-, v_{d-1}^+] \subseteq [z_{d-1}^-, z_{d-1}^+] \\ z_k^- \leq f_1^s(\vec{V}_k) * g_1^s + \dots + f_m^s(\vec{V}_k) * g_m^s \leq z_k^+, \quad k=d, \dots, K \end{cases} \quad (2.25)$$

Після проведення вище наведених перетворень, отримали загальну форму задачі параметричної ідентифікації інтервальних моделей динаміки відновлення кутів рухливості суглобів верхніх кінцівок у вигляді ІСНАР для окремої s-ої моделі претендента. Як відомо, ІСНАР (2.25) має багато розв'язків, якщо вона сумісна [103]. Проте, для побудови математичної моделі динаміки достатньо отримати хоча б один розв'язок цієї системи. Для цього зазвичай використовують ітераційну процедуру пошуку розв'язку із такої оптимізаційної задачі: [103]:

$$\delta(\lambda_s) \lambda_s = \left\{ f_1^s(\vec{V}) * g_{l1}^s, \dots, f_{m_s}^s(\vec{V}) * g_{lm_s}^s \right\} \min, \quad (2.26)$$

$$m_s \in [I_{\min}, I_{\max}], \quad f_1^s(\vec{V}), \quad f_{m_s}^s(\vec{V}) \in F, \quad \hat{g}_{jl}^s \in [g_{jl}^-, g_{jl}^+], \quad j=1, \dots, m, \quad l=1, \dots, S$$

де  $m_s$  - кількість структурних елементів s-ої інтервальної моделі,  $[I_{\min}, I_{\max}]$  - мінімальне і максимальне значення кількості структурних елементів в моделі, S - кількість моделей-претендентів, F - множина потенційних структурних елементів моделі,  $\hat{g}_{jl}^s$  - вектор оцінок параметрів моделі-претендента зі структурою  $\lambda_s$ .

Процедура оцінювання розв'язків ІСНАР для фіксованої s-ої моделі-претендента ґрунтується на обчисленні якості оцінки параметрів математичної моделі. Тобто взамін функції мети  $\delta(\lambda_s)$ , обчислюють якість оцінки параметрів за допомогою функції  $\delta(\hat{g}_l)$ , яка є різницею центрів найбільш віддалених між собою прогнозованого та експериментального інтервалів для кожної часової дискрети, якщо ці інтервали не перетинаються. У випадку перетину цих інтервалів, функцію  $\delta(\hat{g}_l)$  визначатимемо найменшою шириною їх перетину. Вираз для функції  $\delta(\hat{g}_l)$  представимо у такому вигляді [103]:

$$\begin{aligned} \delta(\hat{g}_l) &= \max_{i=1, \dots, N} \left\{ \left| \text{mid}([\hat{v}_k]) - \text{mid}([z_k^-; z_k^+]) \right| \right\}, \\ \text{якщо } [\hat{v}_k] \cap [z_k^-; z_k^+] &= \emptyset, \quad \forall k=0, \dots, K \end{aligned} \quad (2.27)$$

$$\delta(\hat{g}_l) = \max_{i=1, \dots, N} \left\{ \text{wid}([\hat{v}_k]) - \text{wid}([\hat{v}_k] \cap [z_k^-; z_k^+]) \right\},$$

$$\text{якщо } [\hat{v}_k] \cap [z_k^-, z_k^+] = \emptyset, \forall k=0, \dots, K \quad (2.28)$$

Як бачимо, функція мети є достатньо складною, нелінійною та заданою алгоритмічно. У цьому випадку, для розв'язування оптимізаційної задачі (2.26) необхідно використовувати метаевристичні алгоритми, зокрема метод на основі поведінкової моделі бджолоїної колонії [127]. Розглянемо основні фази методу [128].

Основна ідея методу полягає в аналогії між пошуком оптимальних параметрів моделі із пошуком бджолоїною колонією джерел з найбільшою кількістю нектару. У контексті поведінки бджолоїної колонії, збір нектару поділяється на фази [90,110]: фаза бджіл розвідників, фаза робочих бджіл і фаза бджіл дослідників. Бджоли-розвідники шукають та оцінюють джерела нектару у випадковому напрямі на початковій фазі. Якість джерела нектару, - це певна кількісна характеристика, яка залежить від кількості нектару та відстані до вулика. Після повернення до вулика бджоли-розвідники обмінюються інформацією з робочими бджолами. Останні, - обирають джерело нектару до якого полетять. Чим більша цінність джерела нектару, тим більше бджіл до нього полетить. Якщо джерело вичерпалося, знову настає фаза бджіл-розвідників.

В основу алгоритму покладено зміст фаз бджолоїної колонії. Першим кроком алгоритму є введення початкових змінних: інтервальні дані отримані в ході проведення експерименту; параметри алгоритму структурної ідентифікації, такі як  $S$  - кількість моделей-претендентів,  $MCN$  - загальна кількість ітерацій,  $LIMIT$  - число, яке визначає вичерпність джерела,  $[I_{min}, I_{max}]$  - мінімальне і максимальне значення кількості структурних елементів в моделі,  $F$  - множину потенційних структурних елементів моделі; умови зупинки алгоритму, послідовну чи паралельну стратегію виконання, та нижню і верхню межу можливих значень вектора оцінок параметрів моделі-претендента.

На фазі ініціалізації формуємо початкову множину моделей-претендентів  $A_0$  потужністю  $S$  випадковим чином із набору структурних елементів  $F$ . За допомогою алгоритму параметричної ідентифікації для кожної моделі обчислюємо значення функції мети.

Фаза робочих бджіл відповідає за синтез множини поточних моделей  $\lambda'_{mch}$ . На основі моделі  $\lambda_s$  формуємо нову модель  $\lambda'_s$ , яка є “околом”. При цьому нову модель  $\lambda'_s$  формуємо у спосіб випадкового вибору і заміни частини структурних елементів поточної моделі  $\lambda_s$ . Під час обчислення елементів  $n_s$ , які потрібно замінити, враховуємо якість  $\delta(\lambda_s)$  поточної моделі, яку обчислюємо за допомогою функції  $\delta(\hat{g}_l)$  (2.26) чи (2.27) для фіксованої структури моделі і кількості її елементів  $m_s \in [I_{min}, I_{max}]$ . На цій же фазі проводимо селекцію, для вибору кращої моделі з поточної і генерованої.

В контексті задачі структурної ідентифікації фази бджіл дослідників означає визначення околу дослідження поточної моделі  $\lambda_s$ . Для генерації околу використовуємо описаний вище метод. Для визначення кількості моделей в околі будемо використовувати ймовірнісний підхід, який описаний формулою (2.29):

$$P_s(\lambda_s^1) = \frac{1 - \delta(\lambda_s^1)}{\sum_{s=1}^S (1 - \delta(\lambda_s^1))}, \quad s=1..S \quad (2.29)$$

Отримуємо точні значення кількості новостворених моделей в околі  $m_s = \int (P_s * S)$ . Для кожної моделі введемо показник limit, що відповідає за вичерпність джерела нектару в контексті поведінкової моделі бджолоїної колоній, і слугує умовою виходу із локальних мінімумів в контексті розв'язання оптимізаційної задачі структурної ідентифікації. Далі на цій фазі проводимо групову селекцію для кожного сформованого околу поточної моделі  $\lambda_s$ . Значення лічильника limit збільшуємо на одиницю кожного разу якщо при груповій селекції поточна модель не оновилася.

Наступним кроком обираємо модель з найменшим значенням функції мети  $\delta(\lambda_s)$ . Якщо  $\delta(\lambda_s) = 0$ , завершуємо процедуру структурної ідентифікації, в протилежному випадку переходимо до наступної фази.

Фаза бджіл розвідників реалізовує механізм виходу з локальних мінімумів. Якщо значення лічильника limit поточної моделі  $\lambda_s$  перевищило значення змінної LIMIT, вказаної на етапі ініціалізації, тоді ця модель вважається локальним мінімумом і її потрібно замінити. Нову модель генеруємо випадковим чином, так

само як на фазі ініціалізації.

Як вже зазначалося, під час зміни набору структурних елементів для кожної моделі претендента, здійснюємо пошук оптимальних оцінок параметрів за допомогою цього ж самого метаевристичного алгоритму. Розглянемо деталі цього ж алгоритму.

Перед початком роботи алгоритму задаємо вхідні дані та параметри алгоритму: інтервальні дані (1); структуру моделі (2.19);  $S$  - чисельність усієї популяції бджіл;  $M_{CN}$  - загальну кількість ітерацій,  $m_{cn}=1$  - номер поточної ітерації,  $LIMIT$  - числове значення, яке визначає вичерпність джерела.

Фаза ініціалізації (бджіл розвідників). На цій фазі ініціалізуємо вектори  $\hat{g}_l$  випадковим чином, використовуючи таку формулу:

$$[\hat{g}_{jl}^-; \hat{g}_{jl}^+] = [g_{jl}^{low}] + rand(0,1) * (g_{jl}^{up} - g_{jl}^{low}), \quad j=1, \dots, m, \quad l=1, \dots, S \quad (2.30)$$

Далі, обчислюємо значення функції мети за формулою (2.26) чи (2.27). Як бачимо, обчислення значень функції мети незалежні між собою для кожного із згенерованих за формулою (2.30) векторів  $\hat{g}_l$ . Тому ці обчислення доцільно проводити в паралельних потоках. Якщо кількість обчислювальних вузлів  $N$ , то усю множину векторів  $\hat{g}_l, l=1, \dots, S$  розділимо на  $N$  частин.

Фаза робочих бджіл. В контексті оптимізаційної задачі ця фаза означає пошук нових оцінок розв'язків, які забезпечують зменшення значення функцію мети. Такі оцінки вектора параметрів обчислюємо за формулою:

$$\hat{g}_{jl}^{mcn} = \hat{g}_{jl} + \Phi_{jl} * (\hat{g}_{jl} - \hat{g}_{jp}), \quad j=1, \dots, m, \quad p \neq l=1, \dots, S \quad (2.31)$$

де  $\Phi_{jl}$  - випадкове число з діапазону  $[-1;1]$ ,  $j=1, \dots, m$  - випадково обраний індекс параметра,  $\hat{g}_p$  - випадково обраний вектор координат нектару із  $p \neq l=1, \dots, S$ .

Якщо отримане значення параметру не задовольняє умові  $\hat{g}_l \in [g_{jl}^{low}; g_{jl}^{up}]$ , то замість формули (2.31) використовуємо таку формулу:

$$\hat{g}_{jl}^{mcn} = \hat{g}_{jl} - \Phi_{jl} * (\hat{g}_{jl} - \hat{g}_{jp}), \quad j=1, \dots, m, \quad p \neq l=1, \dots, S \quad (2.32)$$

Далі, попарно порівнюємо значення функції мети для поточних та обчислених оцінок параметрів:

$$\hat{g}_i = \left\{ \hat{g}_i, \text{ якщо } \delta(\hat{g}_i) \leq \delta(\hat{g}_i^{mcn}) \right\} \text{ або } \hat{g}_i = \left\{ \hat{g}_i^{mcn}, \text{ якщо } \delta(\hat{g}_i) > \delta(\hat{g}_i^{mcn}) \right\} \quad (2.33)$$

Після цього збільшуємо лічильник спроб (вичерпаності джерела нектару)  $\text{limit} < \text{LIMIT}$  для отриманого значення  $\hat{g}_i$ , щоб мати можливість вийти з локального мінімуму в майбутньому.

Як бачимо, обчислення значень функції мети незалежні між собою для кожного із згенерованих за формулами (2.31), (2.32) векторів  $\hat{g}_i$ . Тому ці обчислення також доцільно проводити в паралельних потоках.

Фаза бджіл дослідників. В контексті оптимізаційної задачі, на цій стадії визначаємо найбільш ймовірні точки (вектори значень параметрів), в околі яких необхідно проводити детальне дослідження функції мети. Для цього для кожної точки, встановленої на попередній фазі, обчислюємо ймовірність  $P_i$  за наступною формулою:

$$P_i = \frac{1 - \delta(\hat{g}_i)}{\sum_{l=1}^s (1 - \delta(\hat{g}_l))} \quad (2.34)$$

Варто зазначити, що попередньо значення  $\delta(\hat{g}_i)$  нормуємо до 1.

На підставі розрахованих ймовірностей за формулою (2.34) визначаємо кількість точок для дослідження околу можливих локальних мінімумів функції мети -  $m_i = \int (P_i * S)$  точок. Відповідно у випадку  $m_i > 0$ , генеруємо  $m_i$  сусідніх точок за формулами (2.31) чи (2.32). Після обчислення координат кожної точки околу, організуємо  $N$  потоків обчислень, подібно, як на попередніх фазах. В кожному потоці для точок, в околі яких  $m_i > 0$ , обчислюємо значення функції мети і попарно порівнюємо ці значення для поточних та обчислених оцінок параметрів за формулою (2.33).

Наступний кроком є пошук точки з найменшим значенням функції мети на даній ітерації алгоритму. Якщо значення функції мети  $\delta(\hat{g}_i) = 0$ , то завершення процедури оптимізації. В протилежному випадку - покладемо  $\text{limit} = 0$  і переходимо на початок цієї фази.

Фаза бджіл розвідників. Ця фаза використовується для уникнення зациклення на локальних мінімумах функції мети. В контексті поведінкової моделі бджолоїної

колонії, на цю фазу переходимо, коли джерело нектару є вичерпаним. Для цього у кожній точці перевіряємо чи не перевищує її лічильник спроб limit граничне значення LIMIT. Якщо- так, то відповідний параметр моделі, замінюємо оцінкою, яку обчислюємо за формулою (2.30). Після цього переходимо до фази робочих бджіл.

Очевидно, що для забезпечення збіжності ітераційної процедури при розв'язуванні задачі (2.25), необхідно забезпечити таку послідовність оцінювання значень функції  $\delta(\hat{g}_1), \dots, \delta(\hat{g}_l)$ , яка забезпечує такі умови [126]:

$$\delta(\hat{g}_1) > \delta(\hat{g}_2) > \dots, \delta(\hat{g}_{l=L} \in \Omega), \quad (2.35)$$

де  $\Omega$  позначає область розв'язків ІСНАР.

### **2.3. Адаптивний цикл прогнозування та корекції терапевтичного протоколу**

Ефективність фізичної реабілітації верхніх кінцівок суттєво залежить від здатності терапевта своєчасно адаптувати терапевтичний протокол до індивідуальної динаміки відновлення конкретного пацієнта [2, 53]. Традиційний підхід до управління реабілітаційним процесом є переважно реактивним: рішення щодо зміни інтенсивності, складності або характеру вправ приймаються на основі ретроспективного аналізу даних після завершення чергового сеансу, що не дозволяє оперативно реагувати на уповільнення прогресу або негативні тенденції у динаміці рухливості [62]. Відсутність формалізованих механізмів прогнозування та формування рекомендацій обмежує потенціал цифрових реабілітаційних платформ, залишаючи прийняття ключових терапевтичних рішень виключно на суб'єктивній оцінці спеціаліста [41, 42].

У даній роботі запропоновано перехід від реактивного до проактивного управління реабілітаційним процесом шляхом побудови адаптивного циклу прогнозування та корекції терапевтичного протоколу. Концепція адаптивного циклу базується на використанні інтервальних математичних моделей динаміки відновлення кутів рухливості суглобів (підрозділ 2.2), результати прогнозування

яких є ключовим інформаційним джерелом для формування рекомендацій щодо корекції терапії. На відміну від систем підтримки прийняття рішень, що ґрунтуються на фіксованих клінічних правилах [112, 113], запропонований підхід формує персоналізовані рекомендації з урахуванням індивідуальної прогнозованої динаміки кожного пацієнта, що є суттєвою перевагою у контексті реабілітації верхніх кінцівок, де індивідуальна варіабельність відновлення є надзвичайно високою [129].

Адаптивний цикл реалізує замкнений контур керування реабілітаційним процесом, що складається з п'яти послідовних етапів:

1. Виконання терапевтичного завдання пацієнтом у AR-середовищі з арт-терапією.

2. Автоматичне вимірювання кутів рухливості суглобів системою безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією (підрозділ 2.1).

3. Побудова інтервальних прогнозних моделей динаміки відновлення на основі накопичених часових рядів (підрозділ 2.2).

4. Аналіз прогнозованої траєкторії та формування рекомендацій щодо корекції терапевтичного протоколу.

5. Реалізація корекції через зміну терапевтичного зображення та параметрів AR-сесії.

Після виконання корекції цикл повторюється: пацієнт виконує наступний сеанс із оновленим терапевтичним зображенням, нові вимірювання збагачують часовий ряд, після чого здійснюється перевірка адекватності поточної інтервальної моделі: фактично виміряне значення кута перевіряється на належність до прогнозного інтервалу, побудованим за моделлю для відповідного сеансу, і модель вважається адекватною, якщо це значення належить розширеному прогнозному інтервалу - за умови адекватності модель зберігається та використовується для рекурентного обчислення прогнозу на наступні сеанси, інакше виконується її повна переідентифікація на оновленому часовому ряді з визначенням нової структури різницевого рівняння та нових оцінок параметрів. Допуск розширення приймається

пропорційним ширині прогнозного інтервалу - такий вибір забезпечує строгіший контроль адекватності для моделей з вищою прогноною визначеністю та обґрунтований у підрозділі 3.4. Такий ітеративний процес забезпечує безперервну адаптацію терапії до фактичної динаміки відновлення. Загальну структуру адаптивного циклу з деталізацією компонентів аналізу, генерації та корекції подано на рисунку 2.3.

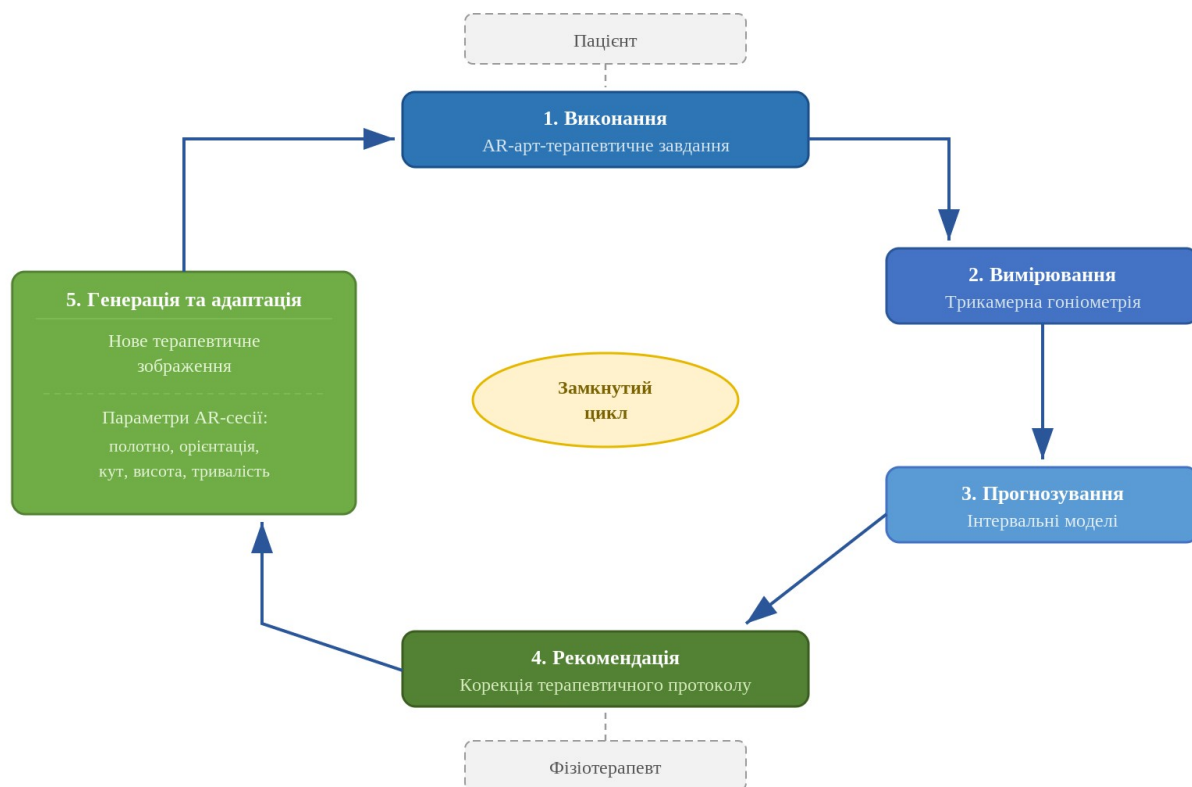


Рис. 2.3. Схема замкнутого адаптивного циклу реабілітації верхніх кінцівок

Центральним елементом адаптивного циклу є компонент аналізу прогнозу, який використовує результати інтервальної математичної моделі динаміки відновлення (підрозділ 2.2). Модель, побудована на основі часового ряду вимірних значень кутів рухливості за попередні сеанси, дозволяє рекурентно обчислювати прогнозовані інтервальні значення кута для кожного наступного сеансу. Таким чином, після проведення достатньої кількості сеансів (мінімум три, відповідно до порядку різницевого рівняння  $d = 3/4$ ) система отримує послідовність прогнозованих інтервалів на майбутні сеанси: четвертий, п'ятий, шостий і так далі.

Кожен прогнозований інтервал містить нижню та верхню межу можливого значення кута, що відображає невизначеність, притаманну процесу відновлення.

Компонент аналізу порівнює ці прогнозовані значення з цільовими показниками, визначеними на основі клінічних стандартів AAOS [124] (згинання плечового суглоба 160-180°, розгинання 50-60°, відведення 180°) та індивідуальних цілей реабілітації конкретного пацієнта. Оскільки модель генерує прогноз для кожного наступного сеансу, система може безпосередньо визначити, на якому саме сеансі прогнозоване значення кута вперше досягне або перевищить цільовий діапазон. Це дає пряму оцінку очікуваної кількості сеансів до завершення реабілітації. Система класифікує стан реабілітаційного процесу на основі порівняння цієї прогнозованої кількості зі стандартною тривалістю курсу (типово 10 сеансів): позитивна динаміка - коли прогнозована кількість не перевищує стандартну; помірне відхилення - коли перевищення становить до 50%; та негативний прогноз - коли перевищення складає більше 50%. У разі виявлення помірного відхилення або негативного прогнозу система ініціює формування рекомендацій щодо корекції терапевтичного протоколу.

Ключовим механізмом корекції терапевтичного протоколу в контексті AR-реабілітації з арт-терапією є зміна терапевтичного зображення-розмальовки, яке пацієнт малює на віртуальному AR-полотні під час сеансу. Просторовий розподіл елементів зображення безпосередньо визначає біомеханічний профіль рухів пацієнта: вертикально орієнтовані елементи (високі дерева, архітектурні споруди, вертикальні абстракції) стимулюють згинання плечового суглоба з піднесенням руки угору; горизонтально орієнтовані елементи (пейзажні панорами, широкі мазки) стимулюють відведення та приведення у горизонтальній площині; дрібні деталі стимулюють точні рухи зап'ястка та пальців [46]. Таким чином, зміна зображення є неінвазивним терапевтичним втручанням: пацієнт продовжує виконувати базове арт-терапевтичне завдання, не усвідомлюючи зміни терапевтичної стратегії, тоді як біомеханічний профіль його рухів суттєво трансформується завдяки іншому просторовому розподілу елементів.

Формування рекомендацій базується на встановленні відповідності між типами рухів верхніх кінцівок та просторовими характеристиками терапевтичного зображення. Для кожного типу руху (згинання, розгинання, відведення, приведення) визначено, яка просторова характеристика зображення найбільш ефективно стимулює цей рух: згинання стимулюється вертикально орієнтованими елементами, відведення - горизонтально орієнтованими, а точні рухи зап'ястка - дрібними деталями. Рекомендація формується як комплексний опис бажаних змін у просторовому розподілі елементів зображення, де акцент на кожному типі елементів визначається пропорційно величині відхилення прогнозу від цільового значення для відповідного кута рухливості. Наприклад, якщо аналіз прогнозу виявив уповільнення динаміки для кута згинання плечового суглоба, рекомендація включає збільшення частки вертикально орієнтованих елементів зображення; аналогічно, при негативному прогнозі відведення рекомендується збільшення горизонтальних елементів.

Додатковим напрямком корекції є налаштування параметрів конфігурації AR-сесії, що дозволяє впливати на амплітуду та напрямок рухів: розмір полотна (малий, середній, великий, дуже великий), орієнтація полотна (горизонтальне або вертикальне), кут нахилу полотна ( $45-150^\circ$ ), висота відносно очей пацієнта ( $-50..+50$  см) та тривалість сесії (5-60 хв). Збільшення розміру полотна та його вертикальна орієнтація стимулюють рухи з більшою амплітудою згинання, тоді як горизонтальне розташування сприяє рухам відведення. Зміна висоти полотна відносно очей впливає на початкове положення руки та діапазон руху згинання/розгинання. Таким чином, рекомендація щодо корекції протоколу формується як комплексний вектор змін, що включає як зміну просторового розподілу елементів зображення, так і корекцію параметрів AR-середовища.

Формування рекомендацій підпорядковується системі обмежень безпеки, що враховує клінічні протоколи та індивідуальні характеристики пацієнта. Для кожного типу руху визначено максимально допустиму амплітуду, що залежить від діагнозу, стадії реабілітації та обмежень, встановлених фізичним терапевтом [34]. Рекомендація щодо збільшення інтенсивності рухів формується лише за умови, що

прогнозоване значення кута з урахуванням запропонованої корекції не перевищує безпечний діапазон. При порушенні цієї умови система обмежує рекомендований приріст амплітуди до безпечного значення. Додатково враховується ширина прогнозного інтервалу як міра невизначеності: при надмірно широкому інтервалі прогнозу система рекомендує консервативну стратегію зі збереженням поточних параметрів протоколу, що запобігає прийняттю рішень на основі ненадійних прогнозів [116]. Такий підхід забезпечує баланс між агресивністю терапії (прискорення відновлення) та безпекою пацієнта (запобігання перевантаженню та повторній травматизації), що є критично важливим аспектом у реабілітації верхніх кінцівок [129].

Порівняно з існуючими підходами до рекомендаційних систем у фізичній реабілітації, запропонований адаптивний цикл має ряд суттєвих відмінностей. По-перше, більшість наявних систем базуються на поточних вимірюваннях або ретроспективному аналізі, тоді як запропонований підхід використовує прогнозовані значення, що дозволяє здійснювати проактивну корекцію до появи негативних наслідків [113]. По-друге, використання інтервальних моделей дозволяє враховувати невизначеність вимірювань та прогнозів, що підвищує надійність рекомендацій порівняно з точковими оцінками [37]. По-третє, механізм корекції через зміну терапевтичного зображення є унікальним для AR-арт-терапевтичних систем і забезпечує неінвазивне втручання, непомітне для пацієнта, що підтримує мотивацію та прихильність до програми реабілітації. По-четверте, інтеграція рекомендаційного модуля безпосередньо у вебзастосунок фізіотерапевта забезпечує нативний доступ до всіх необхідних даних та контекстну релевантність рекомендацій у робочому процесі спеціаліста.

Важливим аспектом запропонованого підходу є багатоступеневий AI-контур реалізації рекомендацій. Після формування рекомендації у вигляді семантичного опису бажаних змін у біомеханічному профілі рухів, текстова рекомендація автоматично трансформується великою мовною моделлю (LLM) у структурований промт для генерації зображення [70]. Мовна модель отримує системний контекст про арт-терапевтичне призначення зображень та клінічну рекомендацію щодо

необхідного біомеханічного профілю рухів, на основі чого генерує текстовий опис візуальних характеристик зображення-розмальовки з урахуванням терапевтичних вимог: орієнтацію елементів, їх розподіл у просторі, рівень деталізації та стиль контурів. Отриманий prompt передається до генеративної моделі, яка створює контурне зображення для використання як терапевтична розмальовка на AR-полотні. Такий підхід дозволяє уникнути жорстко заданих шаблонів арт-завдань та забезпечує адаптацію візуального контенту до індивідуальних потреб реабілітації.

Забезпечення пояснюваності рекомендацій є критично важливим аспектом систем підтримки прийняття рішень у медичній сфері [41, 42]. У запропонованому підході кожна рекомендація супроводжується кількісним обґрунтуванням: числовими показниками прогнозу (прогнозоване значення кута на визначений сеанс, очікувана кількість сеансів до досягнення цільового діапазону, відхилення від стандартного курсу), візуалізацією прогнозних траєкторій та текстовим описом причин формування рекомендації. Це дозволяє фізичному терапевту не лише отримувати рекомендації, а й розуміти логіку їх формування, критично оцінювати їх доцільність та приймати остаточне рішення щодо застосування корекції. Принципово важливо, що система є рекомендаційною, а не автономною: остаточне рішення завжди залишається за спеціалістом [113].

Таким чином, запропонований адаптивний цикл прогнозування та корекції терапевтичного протоколу інтегрує три методи, описані у Розділі 2: метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків (підрозділ 2.1) забезпечує об'єктивні вхідні дані; ідентифікація інтервальних моделей динаміки відновлення на основі алгоритму бджолоїної колонії (підрозділ 2.2) забезпечує прогнозування з урахуванням невизначеності; а критерії аналізу прогнозу та механізм відображення рухових параметрів на просторові характеристики терапевтичного зображення забезпечують генерацію обґрунтованих рекомендацій. Замкнений контур «виконання - вимірювання - прогноз - рекомендація - корекція зображення - адаптація рухів» створює основу для персоналізованої проактивної фізичної реабілітації, у якій терапевтичний протокол безперервно адаптується до прогнозованої динаміки відновлення рухливості суглобів верхніх кінцівок.

Ефективність OneEuro-фільтра у порівнянні з альтернативними методами згладжування траєкторій ключових точок наведено у таблиці 2.4. Для повноти також включено базовий метод з адаптивним фільтром Калмана, апробований у попередній публікації авторів (підрозділ 2.1), який за клінічними показниками тремору та затримки відгуку еквівалентний OneEuro, але має значно вищу обчислювальну складність [47].

Таблиця 2.4.

## Порівняння методів фільтрації траєкторій ключових точок

Метод	Затримка	Реакція на стрибки	Фіз. модель	Складність
Ковзне середнє ( $N=5$ )	2-3 кадри	Погана	Ні	$O(1)$
Медіанний фільтр ( $N=5$ )	2-3 кадри	Добра	Ні	$O(N \log N)$
Експ. згладжування ( $\alpha$ )	$\sim 1/\alpha$ кадрів	Залежить від $\alpha$	Ні	$O(1)$
Калман (CV, без <i>gate</i> )	Залежить від $R$	Погана при малому $R$	Так (CV)	$O(n^3)$
Калман + <i>innovation gate</i>	Мінімальна	Відмінна	Так (CV)	$O(n^3)$
OneEuro ( $f_c$ адаптивне)	Мінімальна	Відмінна	Ні (імпл. CV)	$O(1)$

Класичний фільтр Калмана з постійним  $R$  стикається з дилемою: мале значення  $r$  забезпечує швидку реакцію, але спричиняє тремтіння координат; велике  $r$  забезпечує плавність, але створює затримку при різких рухах. Адаптивний інноваційний вентиль вирішує цю дилему динамічним регулюванням  $r$  залежно від норми нев'язки (формула 2.6): при нерухомій руці  $r = r\_stable = 0,06$  забезпечує агресивне згладжування, при різкому русі  $r$  зменшується квадратично, збільшуючи довіру до вимірювання. OneEuro-фільтр досягає функціонально еквівалентної

поведінки через інший механізм - адаптивну частоту зрізу  $f_c = f_{cmin} + \beta \cdot |\dot{x}|$  (формула 2.12), де мала швидкість руху дає малу  $f_c$  (сильне згладжування), а велика швидкість - велику  $f_c$  (швидкий відгук). Ключова перевага OneEuro - обчислювальна складність  $O(1)$  на одну координату проти  $O(n^3)$  для Калмана з матричною інверсією  $S_k \in \mathbb{R}^{2 \times 2}$ , що при 102 скалярних фільтрах сумарно скорочує час фільтрації з 0,5-1,0 мс до  $< 0,1$  мс на кадр, вивільняючи обчислювальні ресурси CPU для інших стадій конвеєра [47].

## Висновки до розділу 2

Розроблено метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією та поєднанні нейромережових детекторів YOLOv8-pose та MediaPipe Hands Landmarker. Запропоновані методи адаптивної фільтрації координат ключових анатомічних точок, зокрема адаптивний фільтр Калмана з інноваційним вентилем та OneEuro-фільтр з адаптивно-керованою частотою зрізу, забезпечують одночасно тремор менше 1 пікселя RMS у стаціонарному режимі та відгук за 1 кадр при різких рухах, що є недосяжним для класичних фільтрів із фіксованими параметрами. Алгоритм геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху зменшує проєкційну похибку з  $30-50^\circ$  до  $3-5^\circ$ , забезпечуючи середню абсолютну похибку менше  $4^\circ$  для всіх типів рухів.

Сформульовано задачі структурної та параметричної ідентифікації інтервальних дискретних динамічних моделей відновлення кутів рухливості суглобів верхніх кінцівок у вигляді різницевого рівнянь. Для розв'язання отриманої інтервальної системи нелінійних алгебричних рівнянь застосовано метаевристичний алгоритм на основі поведінкової моделі бджолиної колонії, що включає фази робочих бджіл, бджіл дослідників та бджіл розвідників із механізмом виходу з локальних мінімумів. Інтервальний підхід дозволяє враховувати невизначеність вимірювань та представляти прогнозовані значення у вигляді інтервалів можливих значень.

Запропоновано адаптивний цикл прогнозування та корекції терапевтичного протоколу, що реалізує замкнений контур керування реабілітаційним процесом

«виконання - вимірювання - прогноз - рекомендація - корекція зображення - адаптація рухів». Розроблено критерії аналізу прогнозованої динаміки для класифікації стану реабілітаційного процесу та механізм формування рекомендацій на основі відповідності між типами рухів верхніх кінцівок та просторовими характеристиками терапевтичного зображення, що забезпечує неінвазивну корекцію біомеханічного профілю рухів пацієнта через зміну арт-терапевтичного контенту.

### РОЗДІЛ 3

## ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ AR-СИСТЕМИ АРТ-ТЕРАПІЇ

У попередньому розділі описано математичне забезпечення комплексної системи: метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків, метод ідентифікації інтервальних моделей динаміки відновлення на основі алгоритму бджолиної колонії та адаптивний цикл прогнозування і корекції терапевтичного протоколу. Сформульовано математичний апарат, що забезпечує перехід від відеоданих до прогнозних інтервалів кутів рухливості суглобів та формування рекомендацій щодо корекції терапевтичного процесу.

У даному розділі описано архітектуру програмної системи реабілітації верхніх кінцівок, яка реалізує розроблені математичні моделі та методи у вигляді комплексного програмно-апаратного рішення для клінічної реабілітації використовуючи AR технологію та арт-терапію. Розглянуто мікросервісну архітектуру системи, побудовану на принципі оркестрації мікросервісів та потоки даних між ними. Описано програмну архітектуру мікросервісу автоматичної гоніометрії на основі аналізу відеопотоків з трьох стаціонарних IP-камер та конвеєрною архітектурою детекції, фільтрації та обчислення кутів.

Далі розглянуто архітектуру AR-застосунку арт-терапії для гарнітури Magic Leap 2 на рушії Unity із трьома ключовими модулями: просторовим позиціонуванням робочої зони, тактильною компенсацією при взаємодії з голографічними об'єктами та рендерингом текстур малювання, а також інтеграцію з серверною частиною через REST API та WebRTC-трансляцію відеопотоку.

Завершальною частиною розділу є опис проектних рішень серверної частини на NestJS із модулями управління пацієнтами, сесіями, WebRTC-сигналізацією та інтервальним прогнозуванням, вебзастосунку фізіотерапевта на React із трьома екранами робочого циклу, а також архітектури рекомендаційної системи підтримки фізичної реабілітації, що замикає адаптивний цикл «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів».

Результати цього розділу опубліковано автором у працях [45, 48, 49].

### **3.1. Загальна архітектура системи та потоки даних між компонентами**

Розроблена система є комплексним програмно-апаратним рішенням для клінічної AR-реабілітації верхніх кінцівок, що функціонує у стаціонарних умовах реабілітаційного кабінету під безпосереднім наглядом фізіотерапевта. Система розвиває попередньо апробовані авторами компоненти: AR-застосунок арт-терапії на Magic Leap 2 [45] та мікросервіс автоматичного гоніометричного аналізу на нейронних мережах [21, 46], інтегруючи їх у єдину архітектуру автоматичної гоніометрії, модулем інтервального прогнозування (C++-адаптер до DLL бібліотеки інтервальної ідентифікації математичних моделей) та автономним веб-клієнтом для незалежного гоніометричного моніторингу. Архітектурне рішення про об'єднання цих компонентів в єдину систему обумовлено необхідністю замкнути адаптивний цикл «виконання -вимірювання -прогноз -рекомендація -корекція» (підрозділ 2.3) у рамках єдиного програмного комплексу, де всі мікросервіси та застосунки обмінюються даними у режимі реального часу без участі оператора.

Система складається з п'яти взаємопов'язаних компонентів: Python-мікросервіс безмаркерного відеоаналізу (Video Analysis Module) на базі трьох стаціонарних IP-камер; AR-застосунку арт-терапії (AR Therapy Application) для гарнітури Magic Leap 2 на рушії Unity; центральний сервервер (Server) на NestJS/TypeScript з базою даних PostgreSQL, що виконує роль API Gateway та оркестратора для усіх інших мікросервісів; мікросервіс інтервального прогнозування (Interval Forecasting Module) - програмного адаптера на C++, що інкапсулює виклики до нативної динамічної бібліотеки (DLL) інтервальної ідентифікації для побудови та використання інтер апропонована архітектура побудована на принципах модульності, розділення відповідальностей та мінімізації зв вальних математичних моделей динаміки відновлення кутів плечового суглоба за алгоритмом штучної бджолоїної колонії; та вебзастосунку фізіотерапевта (Therapist Web Application) на React. Усі компоненти розміщено в локальній мережі реабілітаційного кабінету (Clinic Local Network); мікросервісу вимірювання, AR-

терапії та серверна частина функціонують повністю у локальній мережі без зовнішнього інтернет-з'єднання; зовнішній доступ використовується лише рекомендаційним модулем для генерації терапевтичних зображень через API генеративної моделі, що забезпечує конфіденційність клінічних даних пацієнтів.

Загальну архітектуру системи наведено на рис. 3.1.

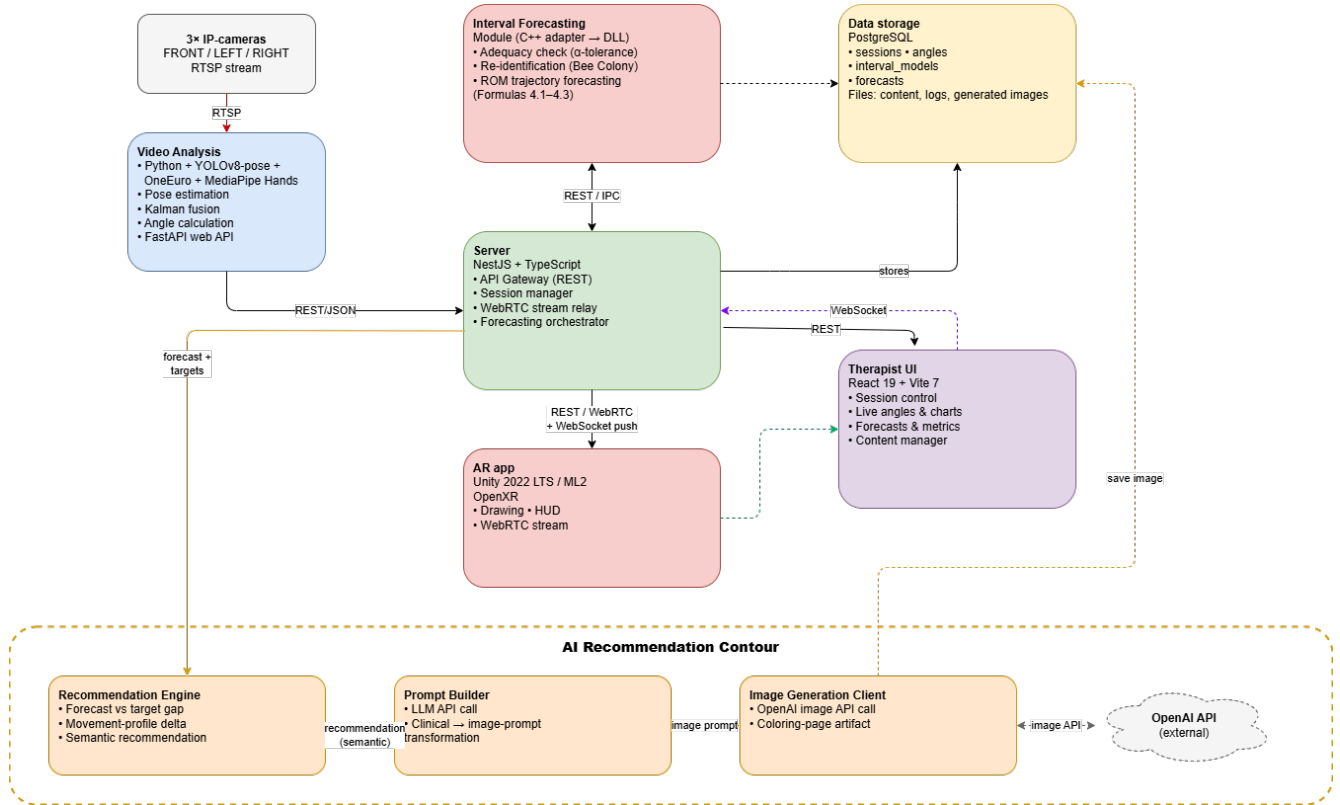


Рис. 3.1. Загальна архітектура комплексної системи AR-реабілітації верхніх кінцівок

Загальний цикл клінічного сеансу організований наступним чином. На підготовчому етапі фізіотерапевт через вебзастосунок реєструє пацієнта або обирає наявну картку, вводить діагноз та визначає початковий протокол реабілітації. Далі фізіотерапевт ініціює сесію та налаштовує параметри AR-сцени: оточення, геометрію полотна, кут нахилу та висоту відносно рівня очей пацієнта, а також завантажує терапевтичне зображення-розмальовку. Пацієнт надягає гарнітуру Magic Leap 2 і виконує арт-терапевтичне завдання -малювання у AR-просторі жестами руки на віртуальному полотні без фізичних контролерів.

Паралельно три IP-камери у реальному часі фіксують рухи кінцівок, Python-мікросервіс обчислює кути рухливості суглобів із застосуванням адаптивної фільтрації координат ключових точок (підрозділ 2.1) та передає їх через NestJS API на вебпанель фізіотерапевта з частотою  $\approx 30$  Гц. Фізіотерапевт спостерігає за кінематичними даними та відеотрансляцією з гарнітури через WebRTC і оперативно коригує параметри сесії - зміни негайно відображаються у гарнітурі без перезапуску AR-застосунку. Після завершення сесії накопичені часові ряди кутових вимірювань автоматично зберігаються у базі даних PostgreSQL, після чого серверна частина ініціює виклик C++-адаптера інтервального прогнозування, який - залежно від адекватності поточної моделі (підрозділ 2.3) - або обчислює прогнозні траєкторії за наявною моделлю, або делегує нативній DLL переідентифікацію інтервальних математичних моделей динаміки відновлення (підрозділ 2.2) із подальшим формуванням прогнозних траєкторій кутів на наступні сеанси; рекомендаційний модуль формує пропозиції щодо корекції наступного терапевтичного зображення на основі цих прогнозів.

У клінічних сценаріях, де AR-гарнітура недоступна або не є необхідною - наприклад, при первинній оцінці рухливості суглобів перед призначенням курсу реабілітації, при тестуванні та налаштуванні системи, або при проведенні реабілітаційних вправ без AR-компоненти, Python-мікросервіс може функціонувати у повністю автономному режимі через вбудований веб-клієнт (vision\_web\_api.py). У такому режимі пацієнт виконує рухи перед трьома камерами без гарнітури, а фізіотерапевт або технічний фахівець спостерігає за кінематичними показниками, станом калібрування та відеопотоками через браузер на будь-якому пристрої в локальній мережі без встановлення додаткового програмного забезпечення. Цей компонент також використовується для первинного налаштування системи: перевірки підключення камер, ручного запуску та зупинки калібрування фільтрів згладжування та верифікації точності вимірювань перед початком клінічного використання.

Взаємодія між компонентами організована через чотири незалежних канали даних із різними протоколами та частотами оновлення. Канал гоніометричних

вимірювань забезпечує передачу кутових даних від Python-мікросервісу через REST API сервера (POST /api/v1/angles, формат JSON, частота  $\approx 30$  Гц) із збереженням у базі даних PostgreSQL; кожен пакет містить мітку часу, ідентифікатор сесії, масив з 22 кутовими значеннями та показник впевненості для кожного параметра. Канал управління AR-сесією передає конфігураційні параметри та контент від вебзастосунку через REST API (PATCH /api/v1/sessions/:id) та WebSocket push-повідомлення через Socket.IO до гарнітури Magic Leap 2, що забезпечує негайне відображення змін у гарнітурі без перезапуску AR-застосунку; затримка між зміною параметра фізіотерапевтом та відображенням у гарнітурі не перевищує 50 мс в умовах локальної мережі. Канал відеотрансляції реалізовано через WebRTC P2P з'єднання між гарнітурою та браузером фізіотерапевта; сервер виконує виключно роль сигнального шлюзу через RoomService, після чого медіатрафік передається безпосередньо з затримкою близько 60 мс. Канал прогнозування забезпечує обробку накопичених часових рядів Interval Forecasting Module на сервері після завершення кожної сесії; сформовані прогнози інтервали та метрики прогресу передаються на вебпанель фізіотерапевта через GET /api/v1/patients/:id/forecast для аналізу та формування рекомендацій.

Запропонована архітектура побудована на принципах мікросервісної архітектури: модульності, розділення відповідальностей, мінімізації зв'язності та незалежного розгортання кожного компонента (таблиця 3.1). Кожний мікросервіс може розвиватися та оновлюватися незалежно за умови збереження контрактів API. Python-мікросервіс комунікує з NestJS-сервером виключно через REST API (POST /api/v1/angles); AR-застосунок отримує конфігурацію через REST та WebSocket; Interval Forecasting Module реалізований як C++-адаптер до нативної DLL інтервальної ідентифікації та викликається серверною частиною через IPC-інтерфейс, що ізолює деталі взаємодії з нативною бібліотекою від решти системи та дозволяє у майбутньому замінити обчислювальне ядро на альтернативну реалізацію без впливу на серверну частину; вебзастосунок взаємодіє з усіма компонентами через серверну частину NestJS, що виконує роль єдиної точки

інтеграції. Веб-клієнт автономного моніторингу взаємодіє безпосередньо з Python-мікросервісом через її внутрішній FastAPI-сервер, що дозволяє використовувати мікросервісу вимірювань без розгортання NestJS-сервера (Додаток В). Такий підхід забезпечує переносимість системи: для розгортання у новому реабілітаційному кабінеті достатньо налаштувати локальну мережу та IP-адреси камер без модифікації програмного коду.

Таблиця 3.1.

## Зведена характеристика архітектури

Мікросервіс/ Застосунок	Технологія	Інтерфейс	Роль у сесії	Виходи даних
Video Analysis Module (Python)	YOLOv8-pose + OneEuro + MediaPipe Hands + PyQt5	RTSP/TCP (вхід), REST/JSON (вихід)	Безмаркерна гоніометрія	JSON кути (22 параметрів)
AR Therapy App (Unity / ML2)	Unity 2022 LTS + OpenXR	REST, WebRTC	AR арт-терапія	Відеопотік H.264
Server (NestJS)	NestJS + TypeScript + PostgreSQL	REST API, WebSocket	Зберігання, оркестрація прогнозування , сигналізація	sessions, roomId, forecasts
Interval Forecasting Module	C++17 (адаптер) + DLL інтервальної ідентифікації	REST / IPC (через NestJS)	Структурна та параметрична ідентифікація інтервальних моделей	Прогнозні ROM- траєкторії, метрики прогресу
Therapist Web App (React)	React 19 + Vite 7	REST, WebSocket, WebRTC	Моніторинг, управління, рекомендації	Конфігурація сесії, контент

Архітектура також передбачає можливість розширення: додавання нових типів сенсорів (IMU, датчики сили), заміна AR-гарнітури (підтримка OpenXR

забезпечує сумісність з іншими пристроями) або інтеграція з електронними медичними записами через HL7 FHIR API для передачі ROM-вимірювань у зовнішні медичні інформаційні системи.

### 3.2. Мікросервіс автоматичної гоніометрії

Мікросервіс автоматичної гоніометрії (Video Analysis Module) є джерелом об'єктивних кінематичних даних для всього програмного комплексу та виконує роль первинного вимірювального інструменту системи. У підрозділі 2.1 детально описано математичний апарат мікросервісу: адаптивний фільтр Калмана з інноваційним вентиляем (формули 2.5-2.7) та його обчислювально оптимізована альтернатива у вигляді OneEuro-фільтра (формули 2.11-2.12), векторний метод обчислення кутів (2.8-2.9) та алгоритм детермінованого вибору камери з аргмакс-правилом (2.13). У даному підрозділі описано програмну архітектуру мікросервісу: модульну структуру, взаємозв'язки між компонентами, організацію багатопотокової обробки відеопотоків, пайплайн обробки одного кадру та веб-інтерфейс незалежного гоніометричного моніторингу.

Мікросервіс реалізовано мовою Python і структуровано як набір із одинадцяти спеціалізованих модулів [47, 48]: `config.py` (централізована конфігурація та параметризація), `joints.py` (анатомічні визначення суглобів та ROM-норми), `core.py` (нейромережева детекція через YOLOv8-pose та MediaPipe Hand Landmarker, а також функція `compute_angles()`), `smoothing.py` (адаптивне згладжування координат ключових точок OneEuro-фільтром), `camera_selector.py` (детермінований вибір камери для кожного суглоба за правилом аргмаксу впевненості з пріоритет-тайм-брейкером), `pipeline.py` (оркестрація паралельних захопних потоків `CaptureThread` × 3, планувальник запусків детекторів зі субсемплінгом `POSE_EVERY_N_FRAMES = 3` та `HANDS_EVERY_N_FRAMES = 4`), `renderer.py` (візуалізація скелету та дуг кутів на відеокадрі), `recorder.py` (`SessionRecorder` для потокової запис сесії у JSON), `main.py` (десктопний графічний інтерфейс PyQt5), `server.py` (FastAPI-сервер з REST-ендпоінтами для доступу до агрегованих результатів), `vision_web_api.py/app.py` (WebRTC-трансляція відеопотоків з MJPEG-фолбеком). Така декомпозиція забезпечує сувору розділеність відповідальностей (таблиця 3.2): зміна параметрів

Таблиця 3.2.

## Зведена характеристика модулів мікросервісу відеоаналізу

Модуль	Основні класи / функції	Залежності	Призначення
config.py	JOINT_RANGES, JOINT_BEST_VIEW, RTSP_CANDIDATES	-	Централізована конфігурація системи
joints.py	JointDef, POSE_IDX_YOLO, HAND_IDX, JOINT_DEFS	config	Анатомічні визначення та 22 кутових параметрів
core.py	InferenceWorker, HandTracker, compute_angles()	joints, config, Ultralytics, MediaPipe	Нейромережева детекція (YOLOv8-pose + MediaPipe Hands)
smoothing.py	KeypointSmoother, OneEuroFilter	config, NumPy	Адаптивне згладжування координат (OneEuro)
camera_selector.py	CameraSelector, JOINT_BEST_VIEW	joints, config	Вибір камери за $\text{argmax}(\text{conf}) + \text{priority tie-breaker}$
pipeline.py	CaptureThread $\times$ 3, Scheduler, AngleEngine	core, smoothing, camera_selector	Оркестрація: захоплення, планування, обчислення кутів
renderer.py	draw_skeleton(), draw_angles(), розмічання ROM	config, OpenCV, Pillow	Візуалізація скелету та дуг кутів на відеокадрі
recorder.py	SessionRecorder (JSON streaming, export_to_csv)	config	Потокова запис сесії у JSON з покадровою структурою
main.py	MainWindow (PyQt5), CalibrationDialog	всі вище + PyQt5	Десктопний GUI
server.py	FastAPIApp, REST-ендпоінти /api/v1/*	pipeline, FastAPI, uvicorn	Серверна частина з REST API
app.py	WebRTCStream, MJPEG fallback	pipeline, aiortc, FastAPI	Відеотрансляція через WebRTC/MJPEG

OneEuro-фільтра не потребує модифікації модуля візуалізації, додавання нових кутових параметрів обмежується змінами у `joints.py` без впливу на решту кодової бази, а заміна нейромережевого детектора потребує змін лише у `core.py` [47].

На рисунку 3.2 представлено діаграму компонентів мікросервісу, що відображає залежності між модулями та зовнішніми бібліотеками. Модуль `config.py` забезпечує централізоване постачання параметрів усім іншим модулям та реалізує патерн «Єдина точка конфігурації» (Single Source of Configuration). Він містить п'ять функціональних груп параметрів: параметри RTSP-підключення (IP-адреси камер у підмережі 192.168.1.0/24, порти, облікові дані з percent-encoding відповідно до RFC 3986), параметри нейромережевих детекторів YOLOv8-pose [17] та MediaPipe Hand Landmarker [131] (пороги достовірності, максимальний розмір кадру для інференції, частота детекції рук), параметри OneEuro-фільтра ( $f_{sample} = 25$  Гц,  $f_{cmin} = 3$  Гц,  $\beta = 0,01$ , значення яких обґрунтовано у підрозділі 2.1), параметри візуалізації (радіуси маркерів, ширина панелі, висота рядка) та параметри логування. Функції `_build_rtsp_url()` та `_build_rtsp_candidates()` формують URL-адреси відеопотоків з підтримкою резервних шляхів (`stream1`, `stream2`) для підвищення відмовостійкості підключення у різних конфігураціях IP-камер [47].

Багатопотокова архітектура мікросервісу побудована за шаблоном `producer-consumer` (рисунок 3.3). Три незалежних `daemon`-потоки `CaptureThread` паралельно обробляють відеопотоки від кожної камери та атомарно оновлюють `cam_frames`; `asyncio event loop` у `pipeline.py` із частотою 30 Гц зчитує `cam_frames` від усіх трьох `Workers` та виконує центральний конвеєр обробки (YOLO-інференс → OneEuro-фільтр → MediaPipe Hands → AngleEngine + CameraSelector); споживачі результатів - GUI-потік `PyQt5` із `QTimer` (33 мс), `FastAPI`-сервер (синхронний `pull` через `/api/v1/angles`), `WebRTC`-стрім (`aiortc peer`-потік з `/stream/{cam_id}`) та `SessionRecorder` (потокова запис у `.json`) - зчитують атомарно оновлений `snapshot` незалежно один від одного.

Tri-Camera Markerless Analysis - Component Diagram

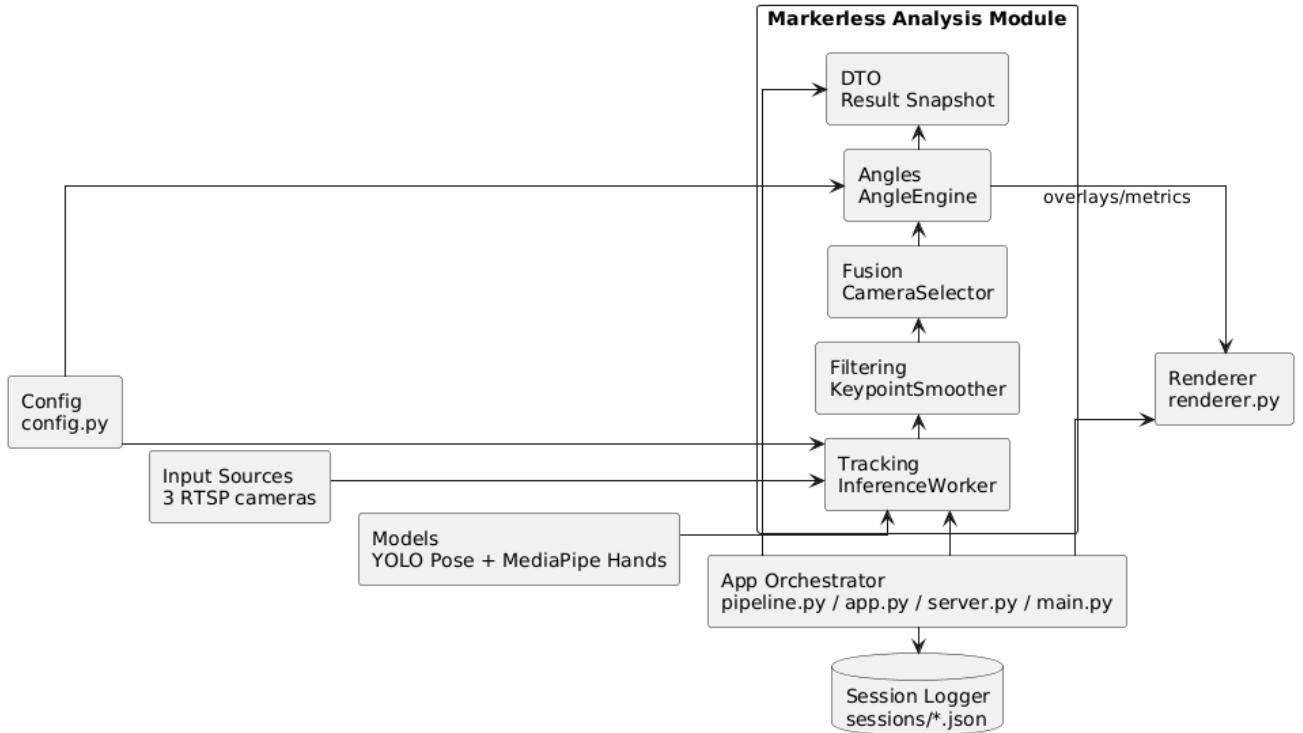


Рис. 3.2. Діаграма компонентів мікросервісу автоматичної гоніометрії

Architecture of Multithreaded Processing - Video Analysis Subsystem

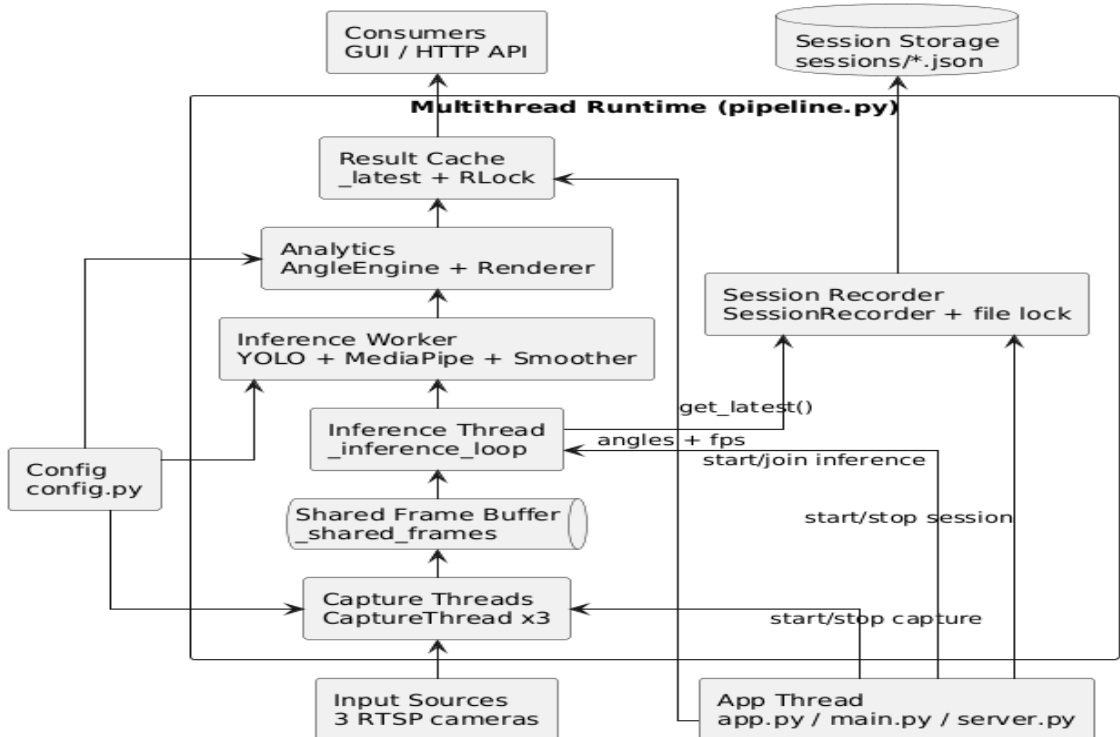


Рис. 3.3. Архітектура багатопотокової обробки мікросервісу відеоаналізу  
Синхронізація між потоками реалізована через threading.Lock із мінімальним часом утримання - операція заміни посилання на словник є  $O(1)$  і не залежить від

обсягу даних. Python GIL гарантує атомарність присвоєння посилань між потоками, а Lock забезпечує видимість змін. Використання даємон-потоків гарантує автоматичне завершення всіх робочих потоків при закритті головного вікна без необхідності явної процедури зупинки [47].

Для детального розкриття послідовності обробки одного відеокадру на рисунку 3.4 наведено розгорнутий пайплайн з дев'яти стадій із вказанням форматів даних на вході та виході кожної стадії.

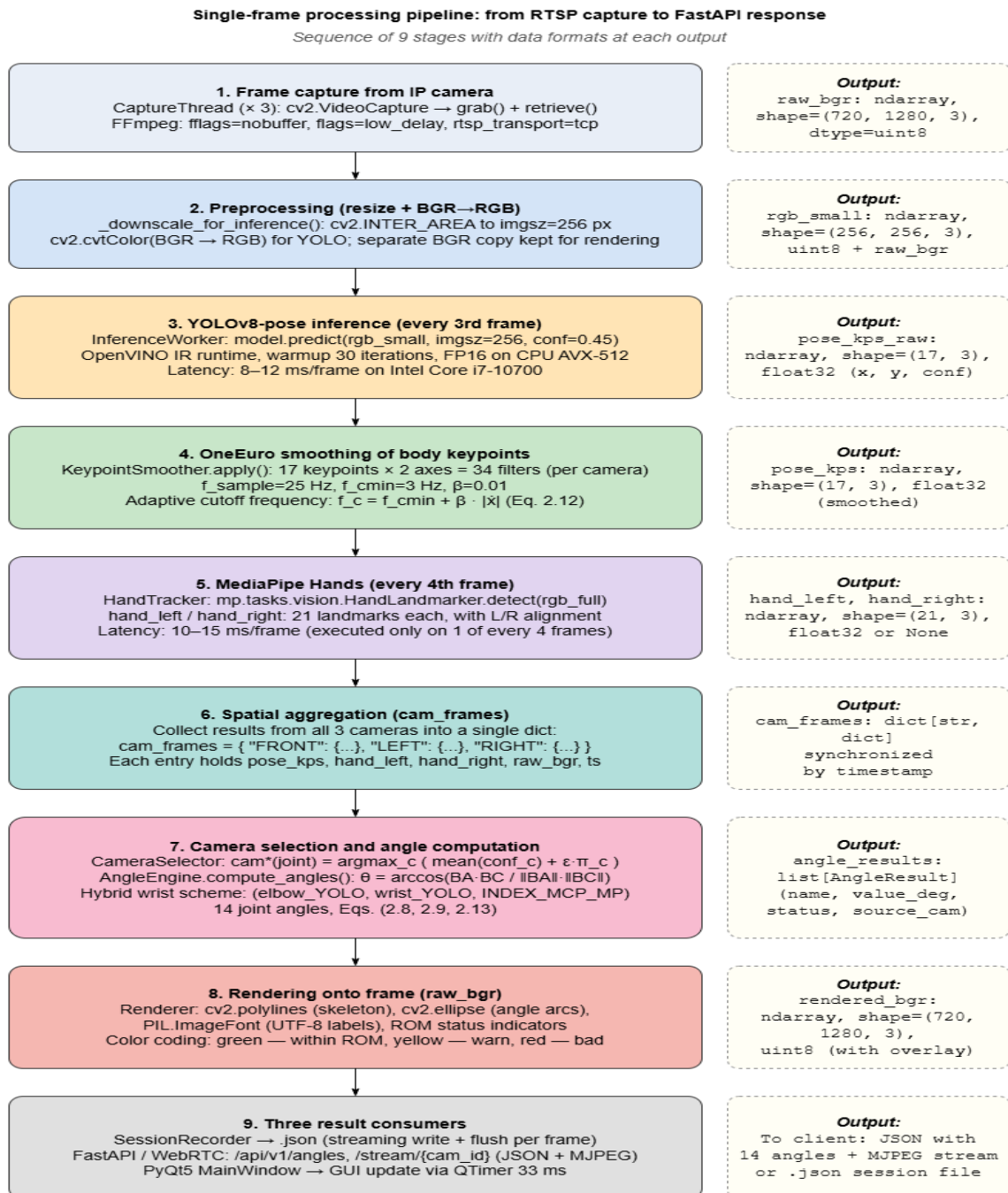


Рис. 3.4. Послідовний пайплайн обробки одного відеокадру з форматами даних

Такий рівень деталізації дозволяє простежити повний шлях перетворення кадру від моменту захоплення з IP-камери через RTSP/TCP до моменту віддачі агрегованого результату трьом споживачам: файлу сесії (SessionRecorder), REST/WebRTC-клієнтам (FastAPI/aiortc) та десктопному інтерфейсу (PyQt5).

На стадії 1 CaptureThread (по одному на кожну з трьох камер) зчитує сирий кадр формату BGR розмірності (720, 1280, 3) через RTSP/TCP з увімкненими FFmpeg-прапорцями мінімальної затримки. На стадії 2 кадр масштабується методом cv2.INTER\_AREA до цільового розміру  $imgsz = 256$  px з конвертацією BGR  $\rightarrow$  RGB для інференсу, паралельно зберігається BGR-копія оригінального кадру для подальшого рендерингу. Стадія 3 - нейромеревий інференс InferenceWorker через YOLOv8-pose з OpenVINO-runtime, виконується на кожному третьому кадрі (POSE\_EVERY\_N\_FRAMES = 3) із середньою латентністю 8-12 мс на процесорі Intel Core i7-10700; результатом є масив (17, 3) - 17 ключових точок COCO з полями  $x, y, conf$ . На стадії 4 KeypointSmoother.apply() згладжує координати через 102 незалежних скалярних OneEuro-фільтри з адаптивною частотою зрізу (формула 2.12). Стадія 5 паралельно виконує MediaPipe Hand Landmarker кожен 4-й кадр (HANDS\_EVERY\_N\_FRAMES = 4) для отримання  $21 \times 2$  точок кистей з латентністю 10-15 мс. На стадії 6 виконується просторова агрегація результатів від усіх трьох камер у єдиний словник cam\_frames. Стадія 7 - вибір камери для кожного суглоба через CameraSelector (формула 2.13) та обчислення 22 кутових параметрів у AngleEngine векторним методом (2.8) з гібридною схемою зап'ястних кутів. На стадії 8 Renderer накладає на BGR-кадр скелет (cv2.polylines), дуги кутів (cv2.ellipse) та україномовні підписи через PIL.ImageFont із кольоровим кодуванням ROM-статусу (ok/warn/bad). На завершальній стадії 9 агрегований snapshot одночасно доступний трьом споживачам: SessionRecorder пише покадрово у JSON з flush() після кожного кадру, FastAPI-сервер повертає JSON-відповідь на GET /api/v1/angles та передає MJPEG-потік через /stream/{cam\_id}, а PyQt5 MainWindow оновлює візуальні елементи через QTimer з періодом 33 мс.

Реєстрація кінематичних даних реалізована модулем recorder.py, що містить клас SessionRecorder для потокової записи у форматі JSON. Кожен кадр

записується як окремий JSON-об'єкт із полями: *frame\_idx*, *t\_sec* (час від початку сесії), *fps*, *angles\_deg* (словник із 22 кутовими значеннями), *status* (словник ok/warn/bad для кожного кута) та *source\_cam* (обрана камера для кожного кута - важливо для відстеження перемикачів). Виклик *flush()* після кожного кадру гарантує збереження даних навіть при аварійному завершенні програми. Метадані сесії включають мітки часу початку та завершення, версію ПЗ, параметри OneEuro-фільтра та інформацію про підключені камери. Функція *export\_to\_csv()* зберігає зведення ROM-параметрів (мінімальне, максимальне та діапазонне значення для кожного з 22 кутів) для подальшого використання серверною частиною (підрозділ 3.4) при побудові інтервальних моделей динаміки відновлення (підрозділ 2.2) [50, 132].

### **3.3. Застосунок арт-терапії для гарнітури Magic Leap 2**

Застосунок арт-терапії у доповненій реальності є третім функціональним компонентом програмного комплексу AR-реабілітації, архітектуру якого описано у підрозділі 3.1. Застосунок реалізовано на ігровому рушії Unity (версія 2022.3 LTS) із використанням стандарту OpenXR [132] та є розвитком прототипу, апробованого у попередній роботі [45]. Апаратною платформою виступає гарнітура доповненої реальності Magic Leap 2 (архітектура x86-64), що забезпечує високоточний трекінг середовища та розпізнавання кистей рук (Hand Tracking) у режимі реального часу [71]. Порівняно з прототипом, поточна версія отримала інтеграцію із серверною частиною NestJS (підрозділ 3.4) для динамічного завантаження налаштувань сесії та терапевтичних зображень із вебзастосунку фізіотерапевта, WebRTC-трансляцію відеопотоку для прямого клінічного спостереження, вдосконалений алгоритм тактильної компенсації та виправлення деформації віртуального пензля. Архітектура побудована за модульним принципом і складається з трьох ключових підсистем: *WorkspaceInitializer* (просторове позиціонування), *BrushManager* (реєстрація взаємодії та тактильна компенсація) та *TexturePainter* (рендеринг текстур малювання), а також інтеграційного модуля мережевої комунікації з віддаленим сервером [48].

При роботі з AR-системами у контексті фізичної реабілітації одним із ключових архітектурних завдань є просторове позиціонування робочої зони таким чином, щоб пацієнту з обмеженою рухливістю не доводилося здійснювати надмірні рухи шийним відділом або шукати інтерфейс у кімнаті. За автоматичне позиціонування відповідає модуль `WorkspaceInitializer`. При запуску програми алгоритм ініціює корутину із заданою затримкою (близько 3 секунд), що надає апаратним сенсорам `Magic Leap 2` час на стабілізацію просторового трекінгу та побудову початкової полігональної сітки приміщення (`Spatial Mapping`). Після стабілізації модуль захоплює поточний вектор напрямку погляду користувача – координати та ротацію головної камери – та проектує його на горизонтальну площину, нівелюючи вертикальні кути нахилу голови пацієнта. Далі відбувається трансляція координат головного робочого об'єкта `WorkSpace` вздовж отриманого вектора на фіксовану відстань 0,75 метра від голови пацієнта із застосуванням кватерніонального повороту безпосередньо обличчям до користувача [48].

Критично важливим архітектурним рішенням модуля `WorkspaceInitializer` є корекція зміщення опорної точки (`Pivot Offset`). Робочий об'єкт `WorkSpace` є збірним і містить декілька дочірніх елементів: інтерактивне полотно для малювання (`Canvas`), панель інструментів із кольоровими кнопками та допоміжні елементи інтерфейсу. Геометричний центр збірного об'єкта, як правило, не збігається з центром полотна для малювання. Без корекції зміщення полотно виявилось б зсунутим відносно центру вектора погляду, що порушує ергономічну позу руки при малюванні. Алгоритм розраховує різницю між позицією кореня збірного об'єкта та фактичною позицією центру полотна, після чого коригує фінальну позицію на цю величину. Таке позиціонування гарантує, що саме інтерактивне полотно опиняється у центрі вектора погляду пацієнта – у межах терапевтичного діапазону рухів верхніх кінцівок [48].

Процес відстеження взаємодії пацієнта з елементами AR-інтерфейсу делеговано модулю `BrushManager`, який вирішує фундаментальну проблему відсутності фізичного зворотного зв'язку при взаємодії з голографічними об'єктами. Оскільки AR-голограми не мають фізичного опору, пацієнт не може

тактильно відчуту момент дотику до віртуального полотна. Традиційний підхід до проектування взаємодії через трасування променів (Raycast) зводиться до випускання віртуального променя безпосередньо з кінчика вказівного пальця. Однак у такому випадку виникає «ефект проколювання»: при зміщенні пальця вперед на 1-2 сантиметри крізь голограму – що часто трапляється у пацієнтів з порушенням моторних функцій – точка початку променя опиняється позаду площини полотна, і процес реєстрації дотику розривається [48].

Для вирішення цієї проблеми у модулі `BrushManager` застосовано метод ретроспективного трасування променя. Кінематичні дані зчитуються з підсистеми `XRHandSubsystem`, що є частиною стандарту `OpenXR`, та локалізується просторовий вузол дистальної фаланги вказівного пальця (відповідно до специфікації `XRHandJointID.IndexTip`). Вектор взаємодії формується зі зсувом назад (`rayOffsetBack = 0,05` м) відносно осі напрямку пальця. Такий ретроспективний зсув забезпечує стійку реєстрацію взаємодії навіть у випадках, коли рука пацієнта тремтить або надмірно заглиблюється у площину полотна в межах заданого допуску глибини. Система реєстрації використовує механізм бітових масок шарів (`Layer Masks`) для диференціації потрапляння променя на два потоки пріоритетів: взаємодія з інтерфейсними інструментами (зміна кольору пензля або перехід у режим ластика) на шарі `uiLayer` та безпосередній контакт із полотном для малювання на шарі `drawingLayer`. Такий підхід повністю виключає конфлікти при обробці подій фізичних границь віртуальних об'єктів (колайдерів), що є критичним для пацієнтів із порушенням дрібної моторики [48].

#### **3.4. Центральний сервер-оркестратор з мікросервісом рекомендацій та вебзастосунок фізіотерапевта**

Центральний сервер (`Server`) та вебзастосунок фізіотерапевта (`Therapist Web Application`) разом утворюють центральний інформаційний вузол програмного комплексу AR-реабілітації. У підрозділі 3.1 визначено роль серверної частини як єдиної точки інтеграції, через яку всі мікросервіси обмінюються даними: Python-мікросервіс безмаркерного відеоаналізу (підрозділ 3.2) передає кутові вимірювання

через REST API, AR-застосунок арт-терапії (підрозділ 3.3) отримує конфігурацію сесій через WebSocket, а вебзастосунок відображає кінематичні дані, прогнозні інтервали та рекомендації для фізіотерапевта. У даному підрозділі описано програмну архітектуру обох компонентів (таблиця 3.3): модульну структуру серверної частини, механізми маршрутизації та валідації запитів, WebRTC-сигналізацію, архітектуру вебзастосунку, його компонентну модель, управління станом, інтеграцію з модулем інтервального прогнозування (підрозділ 2.2), а також архітектуру рекомендаційної системи підтримки фізичної реабілітації на основі корекції терапевтичних зображень [49].

Таблиця 3.3.

Зведена характеристика компонентів серверної частини та вебзастосунку

<b>Компонент</b>	<b>Технологія</b>	<b>Основні класи</b>	<b>Призначення</b>
AppModule	NestJS 11	AppModule, main.ts	Кореневий модуль, глобальні провайдери
PatientsModule	NestJS, TypeORM	PatientsController, PatientsService, Patient	Управління картками пацієнтів
SessionsModule	NestJS, TypeORM	SessionsController, SessionsService, Session	Життєвий цикл сесій
ImageModule	NestJS, fs	ImageController, ImageService	Управління медіафайлами
HealthModule	NestJS	HealthController	Моніторинг стану сервера
WebRtcModule	Socket.IO	WebRtcGateway, RoomService	WebRTC-сигналізація
Forecasting	NestJS	ForecastService	Інтервальне прогнозування

Компонент	Технологія	Основні класи	Призначення
PatientOverview	React 19	PatientOverview, TelemetryGrid	Карта пацієнта, графіки
PatientRegistration	React 19	PatientRegistration, OptionGroup	Введення даних пацієнта
SessionConsole	React 19	SessionConsole, OptionGroup	Конфігурація AR- сесії
API Layer	TypeScript	ApiClient, PatientsApi, SessionsApi, AssetsApi	HTTP-запити до сервера

Серверну частину реалізовано мовою TypeScript на фреймворку NestJS [50] за принципом layered architecture з суворим розділенням відповідальності між шарами. Кожен домен предметної галузі відображається у відокремлений NestJS-модуль, що містить три рівні: Controller (HTTP-шар, який приймає запити та повертає відповіді), Service (бізнес-логіка обробки даних) та DTO (Data Transfer Object – контракти даних для валідації вхідних і вихідних повідомлень). Точкою входу є клас AppModule, який агрегує всі дочірні модулі та налаштовує глобальні провайдери. Такий підхід відповідає принципам SOLID: кожен модуль має єдину відповідальність, залежності інвертовано через Dependency Injection контейнер NestJS, а відкритість для розширення забезпечується додаванням нових модулів без модифікації існуючих [48].

Модульна структура серверної частини включає шість функціональних модулів. Кореневий AppModule підключає ConfigModule для централізованого управління змінними середовища (.env-файл), ServeStaticModule для статичної роздачі медіафайлів, а також чотири доменних модулі: PatientsModule, SessionsModule, ImageModule та HealthModule. Модуль PatientsModule відповідає

за управління записами пацієнтів: створення картки, оновлення демографічних даних та діагнозу, отримання списку пацієнтів із фільтрацією. PatientsController оголошує ендпоінти REST API (POST /api/v1/patients, GET /api/v1/patients, GET /api/v1/patients/:id, PATCH /api/v1/patients/:id), а PatientsService інкапсулює бізнес-логіку роботи з сутністю Patient, що зберігається у базі даних PostgreSQL через TypeORM Repository [48].

**ML Backend - Component Diagram**

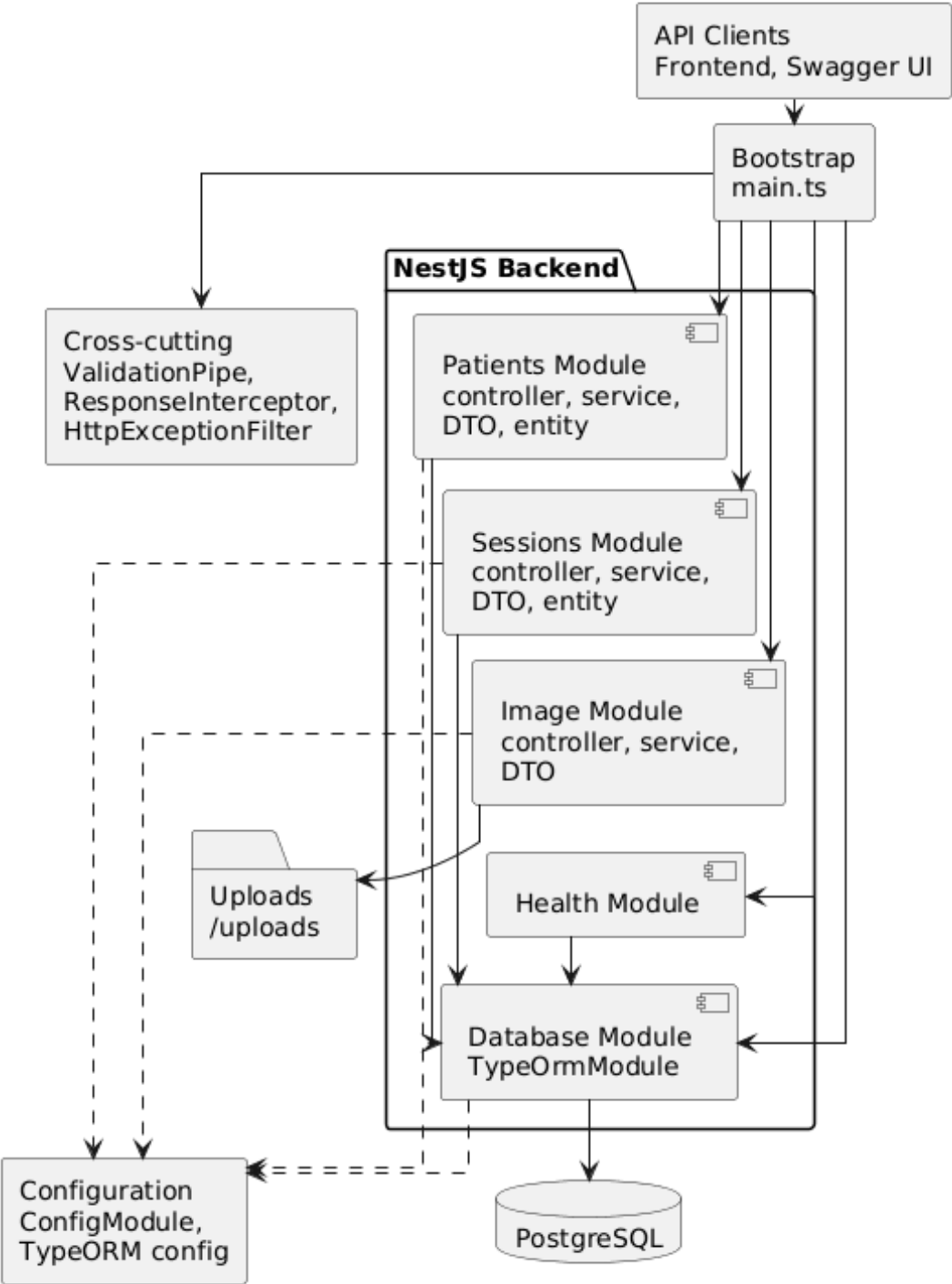


Рис. 3.5. Діаграма компонентів серверної частини NestJS

На рисунку 3.5 представлено діаграму компонентів серверної частини, що відображає залежності між модулями та крос-функціональними провайдерами. Модуль `SessionsModule` управляє життєвим циклом реабілітаційних сесій: створення сесії із прив'язкою до пацієнта, оновлення статусу (PLANNED → ACTIVE → COMPLETED), збереження кутових вимірювань та формування зведених метрик сесії. Ендпоінти REST API (POST `/api/v1/sessions`, GET `/api/v1/sessions`, PATCH `/api/v1/sessions/:id`, POST `/api/v1/sessions/:id/angles`) забезпечують повний CRUD-інтерфейс та прийом потокових даних від Python-мікросервіс відеоаналізу. Модуль `ImageModule` відповідає за управління медіафайлами: формування публічних URL для зображень, перевірку наявності файлів на файловій системі, верифікацію MIME-типів та надання метаданих. Модуль `HealthModule` реалізує моніторинговий ендпоінт GET `/api/v1/health`, що повертає статус сервера, час роботи процесу та мітку часу для інтеграції з системами оркестрації [48].

Усі контракти даних задекларовані з використанням декораторів бібліотек `class-validator` та `nestjs/swagger`, що забезпечує одночасно валідацію вхідних запитів та автоматичну генерацію OpenAPI-специфікації. Наприклад, `CreatePatientDto` містить поля для імені, дати народження, діагнозу та ураженої сторони, кожне з яких анотовано `IsString()`, `IsNotEmpty()` або `IsOptional()` для декларативної валідації. `CreateSessionDto` включає ідентифікатор пацієнта, тип сесії та конфігураційні параметри AR-середовища. Специфікація OpenAPI 3.0 автоматично генерується бібліотекою `@nestjs/swagger` на основі цих декораторів та DTO-класів, забезпечуючи інтерактивну документацію Swagger UI (рисунок 3.6) для розробників та інтеграцію із зовнішніми інструментами генерації клієнтського коду [48].

The image displays the Swagger UI for the 'Video Call & Media API' (OAS 3.0). The interface is organized into several sections, each with a title and a brief description of the endpoints it contains. Each endpoint is listed with its HTTP method, path, and a short description. The endpoints are as follows:

- Images** (Image file serving endpoints):
  - GET /api/v1/images: List all images
  - GET /api/v1/images/ur1: Get image URL by filename
  - GET /api/v1/images/{filename}: Get image URL by path param
- Media** (Monolithic WebSocket media access and signaling endpoints):
  - POST /api/v1/sessions/{sessionId}/media/access: Issue short-lived media access token
  - GET /api/v1/sessions/{sessionId}/media/state: Return current in-memory media debug state
  - POST /api/v1/sessions/{sessionId}/media/access/renew: Renew short-lived media access token
- Patients** (Patient card and overview endpoints):
  - POST /api/v1/patients: Create patient card (intake)
  - GET /api/v1/patients: List patients
  - GET /api/v1/patients/{id}: Get patient by id
  - GET /api/v1/patients/{id}/overview: Get patient overview with latest sessions for frontend dashboard
- forecast** (Interval forecasting for shoulder ROM recovery):
  - GET /api/v1/patients/{id}/forecast: Get interval ROM recovery forecast for a patient
- Sessions** (AR session management endpoints):
  - POST /api/v1/sessions: Create AR therapy session
  - GET /api/v1/sessions: List sessions (optionally filtered by patient)
  - GET /api/v1/sessions/{id}: Get session by id
  - POST /api/v1/sessions/{id}/start: Mark session as active and allow live runtimes
  - POST /api/v1/sessions/{id}/complete: Mark session as completed and stop live runtimes
  - POST /api/v1/sessions/{id}/cancel: Mark session as cancelled and stop live runtimes
- Vision** (3-camera vision worker orchestration and results):
  - GET /api/v1/sessions/{sessionId}/vision: Return current vision analysis state for a session
  - GET /api/v1/sessions/{sessionId}/vision/report: Return final raw vision analysis JSON report
  - GET /api/v1/sessions/{sessionId}/vision/events: Open SSE stream with live vision status, angles, and camera state
  - GET /api/v1/sessions/{sessionId}/vision/streams/{sourceId}: Relay analytics camera stream as MJPEG
- WebRTC** (Session signaling endpoints for WebRTC):
  - POST /api/v1/sessions/{sessionId}/webrtc/join: Join session WebRTC signaling room
  - POST /api/v1/sessions/{sessionId}/webrtc/leave: Leave session WebRTC signaling room
  - GET /api/v1/sessions/{sessionId}/webrtc/participants: List active participants in signaling room
  - POST /api/v1/sessions/{sessionId}/webrtc/offer: Relay WebRTC SDP offer to target participant
  - POST /api/v1/sessions/{sessionId}/webrtc/answer: Relay WebRTC SDP answer to target participant
  - POST /api/v1/sessions/{sessionId}/webrtc/ice-candidate: Relay WebRTC ICE candidate to target participant
  - GET /api/v1/sessions/{sessionId}/webrtc/events: Open SSE stream to receive WebRTC signaling events
- Health** (Health check):
  - GET /api/v1/health: Health check
  - GET /api/v1/health/db: Database health check
- App**:
  - GET /api/v1

Рис. 3.6. Swagger-документація REST API серверної частини системи

На рисунку 3.7 представлено UML-діаграму класів серверної частини. Діаграма демонструє типові патерни NestJS-архітектури: кожен контролер (PatientsController, SessionsController, ImageController) залежить від відповідного сервісу через Dependency Injection; сервіси оперують сутностями TypeORM (Patient, Session) та повертають DTO-об'єкти. Сутність Patient містить поля id (UUID), name, dateOfBirth, diagnosis, affectedSide, createdAt та зв'язок OneToMany з сутністю Session. Сутність Session містить поля id, patientId (зовнішній ключ), status, arConfig (JSON-об'єкт конфігурації AR-сцени), angles (масив кутових вимірювань), createdAt та completedAt. Розділення між сутностями бази даних та DTO-об'єктами для API забезпечує інкапсуляцію внутрішньої структури даних та дозволяє незалежно еволюціонувати схему бази даних та контракти API [48].

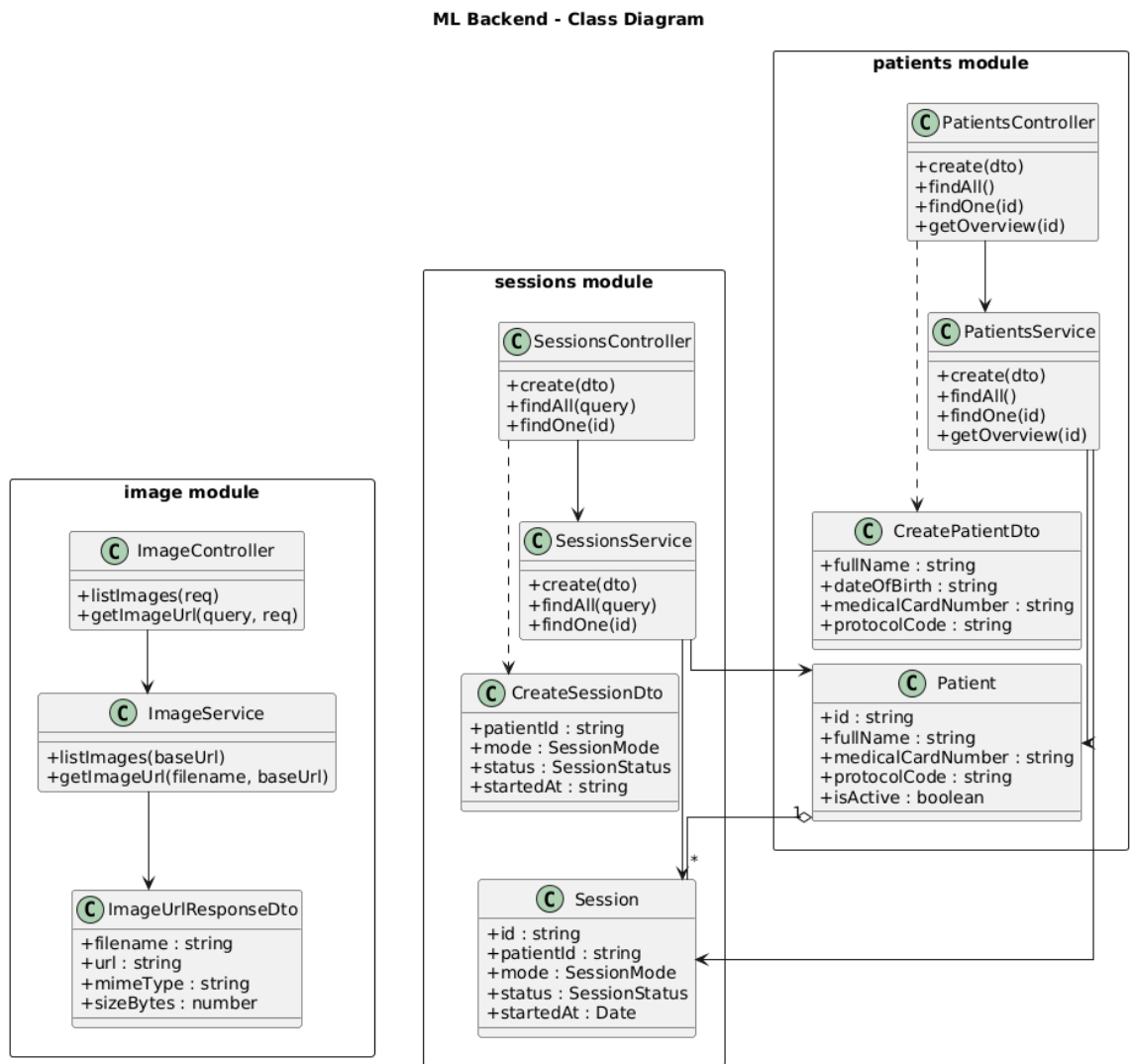


Рис. 3.7. UML-діаграма класів серверної частини NestJS

Персистентне зберігання даних реалізовано через реляційну СУБД PostgreSQL з ORM-бібліотекою TypeORM, що забезпечує декларативне відображення TypeScript-класів на таблиці бази даних (рисунок 3.8). Схема бази даних у поточній версії включає дві основні таблиці: patients (картки пацієнтів) та sessions (терапевтичні сесії). Зв'язок між таблицями реалізовано через зовнішній ключ sessions.patient\_id → patients.id (Many-to-One): один запис у таблиці patients може мати багато записів у sessions, при видаленні пацієнта пов'язані сесії видаляються каскадно [48].

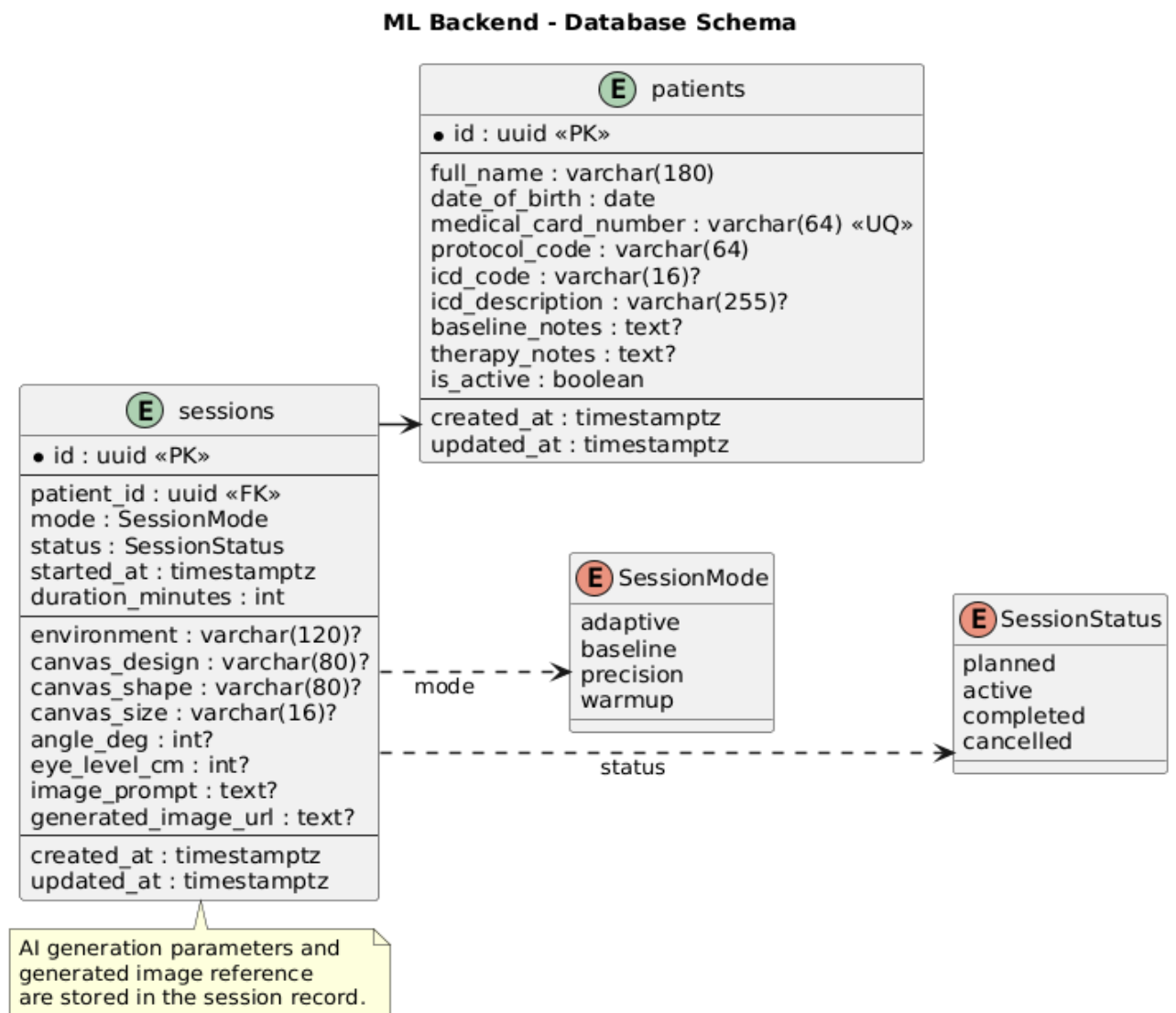


Рис. 3.8. Схема бази даних серверної частини

Таблиця patients зберігає персональні та клінічні дані пацієнта: id (uuid, первинний ключ), full\_name (varchar(180)), date\_of\_birth (date), medical\_card\_number (varchar(64), унікальне обмеження), protocol\_code (varchar(64)

– код реабілітаційного протоколу), `icd_code` та `icd_description` (необов'язкові поля для діагнозу), `baseline_notes` та `therapy_notes` (текстові поля для клінічних приміток), `is_active` (ознака активності пацієнта), а також `created_at` та `updated_at` (мітки часу зі зсувом часової зони). Таблиця `sessions` зберігає інформацію про терапевтичні сесії та параметри генерації AR-сцени: окрім ідентифікатора та зовнішнього ключа `patient_id`, вона містить `mode` (перерахування `SessionMode`: `adaptive`, `baseline`, `precision`, `warmup`), `status` (перерахування `SessionStatus`: `planned`, `active`, `completed`, `cancelled`), `started_at`, `duration_minutes`, а також групу полів конфігурації AR-середовища: `environment`, `canvas_design`, `canvas_shape`, `canvas_size`, `angle_deg`, `eye_level_cm`. Поля `image_prompt` та `generated_image_url` зберігають параметри та результат AI-генерації терапевтичного зображення безпосередньо у записі сесії без використання окремої таблиці для медіа. Файлове сховище для статичних медіаресурсів (шаблони зображень, логотипи AR-сцен) реалізовано через директорію `uploads/` із роздачею через `ServeStaticModule` [48].

Вебзастосунок фізіотерапевта (`Therapist Web Application`) є клієнтською частиною програмного комплексу, що забезпечує повний робочий цикл фізіотерапевта: від ведення картки пацієнта та перегляду кінематичної аналітики до конфігурації та запуску AR-сесій. Застосунок побудований на `React 19` з `TypeScript 5.9` та збиральником `Vite 7` із компілятором `SWC` на `Rust` [48]. Архітектурно вебзастосунок реалізує патерн `Single Page Application (SPA)` із клієнтською маршрутизацією на основі внутрішнього стану `React`. Навігація між екранами реалізована через змінну стану типу `Screen` (`union type` із трьох значень: `overview`, `registration`, `session`), що зменшує складність залежностей та кількість зовнішніх бібліотек. Не використовується жодна зовнішня UI-бібліотека: власна дизайн-система на основі `CSS Custom Properties` забезпечує консистентну візуальну мову інтерфейсу.

На рисунку 3.9 представлено діаграму компонентів вебзастосунку. Архітектура побудована за принципом однонаправленого потоку даних: `Bootstrap`-модуль ініціалізує UI-тему та `App Shell` (компоненти `AppHeader`, `AppSidebar`), який містить `Layout`-компонент із трьома екранами-сторінками. `Шар State Store` управляє

глобальним станом навігації, а Query Layer забезпечує кешування та синхронізацію серверних даних. API Layer (PatientsApi, SessionsApi, AssetsApi) інкапсулює HTTP-запити до серверної частини NestJS через єдиний ApiClient. Такий шаруватий підхід ізолює компоненти інтерфейсу від деталей мережевої взаємодії та дозволяє незалежно тестувати кожен шар [48].

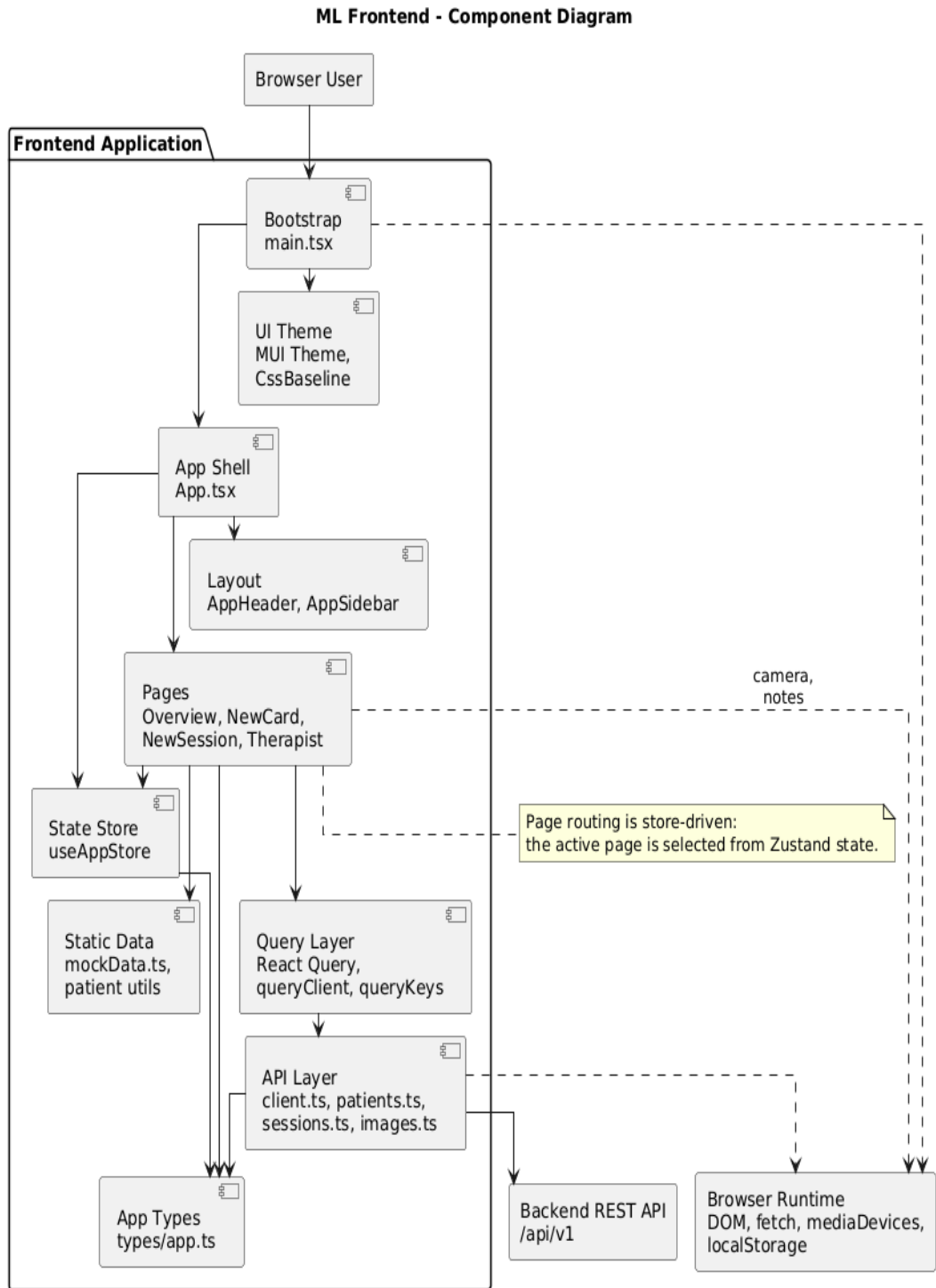


Рис. 3.9. Діаграма компонентів вебзастосунку фізіотерапевта

Екран «Конфігурація AR-сцени» (SessionConsole) є найбільш функціональним екраном системи та дозволяє фізіотерапевту в режимі реального часу налаштовувати параметри AR-середовища для пацієнта перед запуском та під час сесії. Конфігурація включає чотири групи параметрів: вибір оточення (Тренувальний ангар, Суха студія, Нейро-ліс), дизайн полотна (Каркас, Контурний, Моно), форма полотна (Flat, Arc) та розмір (S, M, L, XL). Два слайдери забезпечують точне налаштування геометрії AR-об'єкта: кут нахилу полотна (діапазон 45-150°, крок 15°) та висота відносно рівня очей (-50..+50 см, крок 10 см). Усі параметри формують рядок buildContext, який використовується для AI-генерації терапевтичних зображень-масок. При зміні будь-якого параметра NestJS негайно надсилає push-повідомлення підключеній гарнітурі Magic Leap 2 через Socket.IO без перезапуску AR-застосунку, що є ключовою клінічною функцією адаптивного управління [48].

На рисунку 3.10 представлено UML-діаграму класів вебзастосунку. Діаграма демонструє архітектуру компонентної моделі: App (кореневий компонент) містить AppShell, AppHeader та AppSidebar; PatientOverview, PatientRegistration та SessionConsole є рівноправними сторінками, що рендеряться залежно від стану навігації. AppStore є центральним сховищем стану, що керує змінною screen та даними активного пацієнта. QueryLayer забезпечує кешування серверних даних та автоматичну інвалідацію кешу при мутаціях. Класи PatientsApi, SessionsApi та AssetsApi інкапсулюють типізовані HTTP-запити до відповідних ендпоінтів серверної частини через єдиний ApiClient, що реалізує інтерцептори для додавання заголовків авторизації та обробки мережеских помилок [48].

ML Frontend - Class Diagram

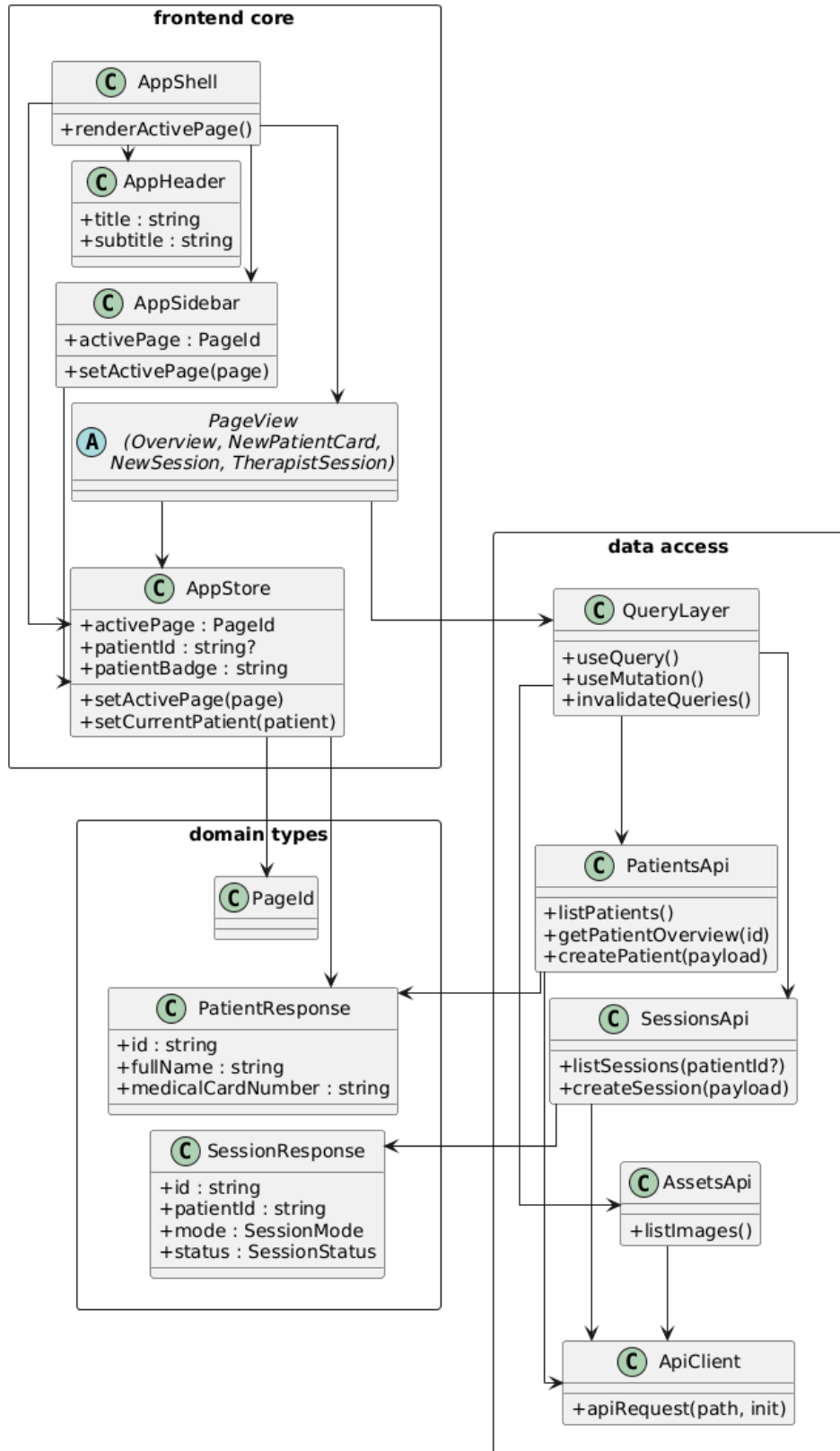


Рис. 3.10. UML-діаграма класів вебзастосунку фізіотерапевта

### Висновки до розділу 3

Спроектовано мікросервісну архітектуру комплексної програмної системи AR-реабілітації верхніх кінцівок, побудовану на принципі оркестрації мікросервісів: мікросервісу аналізу трьох відеопотоків з пріоритизацією на Python, мікросервісу інтервального прогнозування та мікросервісу рекомендацій на основі генеративного штучного інтелекту, оркестрованих центральним сервером на NestJS із базою даних PostgreSQL, а також AR-застосунку арт-терапії для гарнітури Magic Leap 2 на Unity та вебзастосунку фізіотерапевта на React. Основні компоненти розгорнуто в локальній мережі реабілітаційного кабінету; зовнішнє інтернет-з'єднання використовується лише рекомендаційним модулем для генерації терапевтичних зображень. Взаємодію між компонентами організовано через чотири незалежних канали даних: канал гоніометричних вимірювань (REST API, ~30 Гц), канал управління AR-сесією (WebSocket push), канал відеотрансляції (WebRTC P2P, затримка ~60 мс) та канал прогнозування.

Описано програмну архітектуру мікросервісу автоматичної гоніометрії на основі трьох стаціонарних IP-камер з багатопотоковою обробкою відеопотоків. Мікросервіс реалізує конвеєрну архітектуру: детекція анатомічних точок нейромережами YOLOv8-pose та MediaPipe Hands Landmarker, адаптивна фільтрація, геометричне злиття вимірювань від трьох камер та обчислення 22 кутових параметрів суглобів верхніх кінцівок. Три daemon-потоки CameraWorker забезпечують паралельну обробку відеопотоків із частотою ~30 кадрів/с.

Описано архітектуру AR-застосунку арт-терапії на русії Unity 2022.3 LTS із бекендом OpenXR для гарнітури Magic Leap 2. Застосунок побудовано на трьох ключових модулях: WorkspaceInitializer забезпечує автоматичне просторове позиціонування робочої зони на основі вектора погляду пацієнта з корекцією зміщення опорної точки; BrushManager реалізує метод ретроспективного трасування променя для подолання ефекту проколювання голограм із диференціацією взаємодії через бітові маски шарів; TexturePainter виконує прямі маніпуляції з масивом пікселів текстури з автоматичною компенсацією деформації пензля. Інтеграційний модуль забезпечує динамічне завантаження конфігурації та

терапевтичних зображень через REST API, push-оновлення через WebSocket та WebRTC-трансляцію комбінованого відеопотоку з AR-оверлеями.

Розроблено програмну архітектуру серверної частини на фреймворку NestJS за принципом layered architecture із шістьма функціональними модулями та трьома глобальними провайдерами, що уніфікують валідацію, обробку помилок та формат відповідей API. Реалізовано модуль WebRTC-сигналізації для встановлення P2P-з'єднання між гарнітурою та браузером фізіотерапевта, а також модуль інтервального прогнозування, що формує прогнозні інтервали кутів рухливості суглобів на основі ідентифікованих інтервальних моделей. Вебзастосунок фізіотерапевта на React 19 із трьома екранами забезпечує повний робочий цикл: перегляд кінематичної аналітики з прогнозними інтервалами, введення даних пацієнта та конфігурацію AR-сесії в режимі реального часу. Інтегрована рекомендаційна система підтримки використовує прогнозовані інтервальні значення кутів для формування рекомендацій щодо корекції терапевтичного зображення-розмальовки та параметрів AR-середовища через багатоступеневий AI-контур генерації арт-терапевтичного контенту, замикаючи адаптивний цикл «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів».

## РОЗДІЛ 4

### РЕАЛІЗАЦІЯ І АПРОБАЦІЯ AR-СИСТЕМИ АРТ-ТЕРАПІЇ ДЛЯ РЕАБІЛІТАЦІЇ ВЕРХНІХ КІНЦІВОК

У попередньому розділі описано проектування мікросервісної архітектури комплексної програмної системи AR-реабілітації верхніх кінцівок, побудованої на принципі оркестрації мікросервісів: мікросервісу безмаркерного відеоаналізу на базі трьох стаціонарних IP-камер на Python, мікросервісу інтервального прогнозування та мікросервісу рекомендацій на основі генеративного штучного інтелекту, оркестрованих центральним сервером на NestJS із базою даних PostgreSQL, а також AR-застосунку арт-терапії для гарнітури Magic Leap 2 на рушії Unity та вебзастосунку фізіотерапевта на React. Розглянуто загальну архітектуру системи, потоки даних між компонентами та діаграми компонентів кожної мікросервісами та застосунками.

У даному розділі наведено програмну реалізацію всіх компонент комплексу та результати експериментального дослідження розробленої системи в умовах клінічної реабілітації. Спочатку описано реалізацію мікросервісу автоматичної гоніометрії, AR-застосунку арт-терапії та серверної частини з вебзастосунком фізіотерапевта. Далі побудовано інтервальні математичні моделі динаміки відновлення кутів плечового суглоба на основі даних, зібраних під час реабілітаційних сеансів арт-терапії у доповненій реальності з трьома пацієнтами навчальної групи. Для кожного вимірюваного кута виконано структурну та параметричну ідентифікацію із застосуванням алгоритму бджолиної колонії, визначено оптимальні структури моделей та проаналізовано їхню адекватність. Завершальною частиною розділу є апробація адаптивного циклу прогнозування та корекції терапевтичного протоколу на прогнозній групі з двох пацієнтів, де продемонстровано формування негативного прогнозу, автоматичну генерацію рекомендацій та клінічний ефект корекції – перехід від стагнації до стійкого приросту амплітуди рухів.

Результати цього розділу опубліковано автором у працях [45,47,48,49].



числові індекси YOLOv8-pose, що містить 17 точок COCO-keypoints формату. Аналогічно HAND\_IDX визначає 21 точку кисті від зап'ястка до кінчиків пальців для MediaPipe Hand Landmarker. Структура даних JointDef (Python dataclass) описує кожен вимірюваний кутовий параметр: унікальний ідентифікатор (*name*), ключі трьох анатомічних точок (*point\_a*, *vertex*, *point\_b*), нормативні межі ROM за стандартами AAOS [124] (*norm\_min*, *norm\_max*) та метрику обчислення (*metric*: «direct» для геометричного кута, або «from\_straight» для відхилення від прямого положення). Перелік 22 кутових параметрів (по 11 на сторону тіла) охоплює рухи плечового (згинання, розгинання, відведення, приведення, внутрішня та зовнішня ротація), ліктьового (згинання, перерозгинання, пронація, супінація) та зап'ястного суглобів (згинання, розгинання). Для зап'ястних кутів реалізовано гібридну схему з використанням трьох точок з різних джерел: (elbow з YOLO, wrist з YOLO, INDEX\_MCP з MediaPipe Hand Landmarker), що компенсує обмежену точність YOLO для кисті без необхідності повного переключення на детектор кисті [47].

Модуль core.py реалізує нейромережеву детекцію через дві спеціалізовані моделі. Клас InferenceWorker інкапсулює роботу з YOLOv8-pose від Ultralytics: завантажує модель у форматі OpenVINO IR (pose\_n\_openvino\_model/), виконує попередній прогрів (30 ітерацій «порожнього» інференсу для ініціалізації кешу OpenVINO та зменшення латентності перших клінічних кадрів), та надає методи predict(frame) для одиночного інференсу та detect\_batch(frames) для batched-режиму обробки трьох камер одночасно. Модель повертає масив розмірності (17, 3) - 17 ключових точок COCO (*nose*, *eyes*, *ears*, *shoulders*, *elbows*, *wrists*, *hips*, *knees*, *ankles*) з полями *x*, *y* та *conf* (впевненість детекції) [17, 81]. Клас HandTracker інкапсулює MediaPipe Hand Landmarker (Tasks API, модель hand\_landmarker.task) та реалізує алгоритм вирівнювання L/R-мітки рук: оскільки MediaPipe повертає орієнтацію руки відносно зображення, а не відносно пацієнта, для коректного призначення використовуються координати зап'ясть із YOLO-детекції як опорні точки і виконується оптимальне призначення за критерієм мінімальної суми евклідових відстаней між виявленими кистями та відомими позиціями зап'ясть. Для зменшення обчислювального навантаження HandTracker викликається лише на кожному 4-му

кадрі ( $HANDS\_EVERY\_N\_FRAMES = 4$ ), на решті кадрів використовується кешований результат `cached_result`, що є прийнятним з огляду на повільну динаміку зап'ястних кутів у процесі реабілітації. Функція `compute_angles()` будує словник `points_map` із відфільтрованих координат та для кожного визначення з `JOINT_DEFS` обчислює кут векторним методом (формула 2.8) з перетворенням `from_straight` (2.9) для ліктювих та зап'ястних кутів. Результатом є список об'єктів `AngleResult` зі значенням кута, ознакою відповідності ROM-нормі та піксельними координатами для візуалізації [47] (Додаток А).

Модуль `smoothing.py` реалізує клас `KeypointSmoother` - менеджер ансамблю скалярних `OneEuro`-фільтрів, математичний апарат якого описано у підрозділі 2.1 (формули 2.11-2.12). Для кожної координати ( $x$  або  $y$ ) кожної з 17 ключових точок кожної з 3 камер підтримується незалежний екземпляр `OneEuroFilter`, що дає загалом  $17 \times 2 \times 3 = 102$  скалярних фільтрів. Клас `OneEuroFilter` реалізує рекурсивне експоненційне згладжування з адаптивно-керованою частотою зрізу: на кожному виклику обчислюється оцінка швидкості з  $d\_cutoff = 1,0$  Гц, на основі якої визначається адаптивна частота зрізу  $f_c = f_{cmin} + \beta \cdot |\dot{x}|$  та коефіцієнт згладжування  $\alpha_k = 1 / (1 + \tau_k / T_e)$ . Метод `KeypointSmoother.apply(kps_17x3)` приймає сирі координати від `YOLOv8` та повертає згладжений результат тієї ж розмірності, застосовуючи відповідні фільтри для кожної (точки, осі). Метод `reset()` використовується при перепідключенні камери для ініціалізації «холодного старту» фільтрів. Обчислювальна складність однієї ітерації фільтра становить  $O(1)$  на одну координату, що у сукупності дає загальний час фільтрації  $< 0,1$  мс на кадр для всіх 102 фільтрів - на порядок менше порівняно з аналогічним ансамблем фільтрів Калмана [47] (Додаток А).

Модуль `camera_selector.py` реалізує детермінований алгоритм вибору камери для кожного суглоба. Клас `CameraSelector` містить словник `JOINT_BEST_VIEW` - емпірично визначений пріоритетний порядок камер для кожного з 22 кутових параметрів (наприклад, для згинання лівого ліктя пріоритет [`LEFT`, `FRONT`, `RIGHT`], для відведення плеча - [`FRONT`, бічна, протилежна]), що відображає геометрію площини руху. Метод `select(joint, cam_confs)` приймає

словник  $\{cam\_id \rightarrow \text{список conf-значень трьох ключових точок кутового трикутника з відповідної камери}\}$  та повертає  $cam\_id$  за правилом аргмаксу  $cam^*(joint) = \text{argmax}_c (\text{mean}(conf\_c) + \varepsilon \cdot \pi\_c(joint))$ , де  $\text{mean}(conf\_c)$  - середня впевненість детекції трьох ключових точок з камери  $c$ ,  $\pi\_c(joint) \in \{0, 1, 2\}$  - ранг камери у JOINT\_BEST\_VIEW,  $\varepsilon = 0,01$  - коефіцієнт тай-брейкера, який забезпечує детермінований вибір при близьких значеннях confidence (формула 2.13). Ключова перевага такого підходу порівняно з багатоконпонентним зваженим скорингом полягає у повній детермінованості: при однакових вхідних значеннях confidence система завжди обирає одну і ту ж камеру, що забезпечує відтворюваність клінічних результатів та відсутність візуальних стрибків кутових значень між послідовними кадрами. Метод `tie_breaker()` використовується для розв'язання рідкісного випадку, коли впевненості детекції двох камер співпадають до шостого знаку після коми [47].

Центральним модулем мікросервісу є `pipeline.py`, що реалізує клас `Pipeline` - оркестратор усіх обчислювальних ресурсів. При ініціалізації `Pipeline` створює три даемон-потоки `CaptureThread`, по одному на кожну камеру, кожен з яких у замкненому циклі викликає `cv2.VideoCapture.grab() + retrieve()` та атомарно оновлює глобальний словник `cam_frames` записом для відповідної камери, що містить `raw_bgr`-кадр та `timestamp`. Основний `asyncio event loop` з частотою 30 Гц зчитує `cam_frames` від усіх трьох `Workers`, передає кадри у `InferenceWorker.detect_batch()` для `batched`-інференсу YOLO (один раз на 3 кадри згідно з `POSE_EVERY_N_FRAMES`), передає результати у `KeypointSmoother.apply()` для згладжування координат, опціонально викликає `HandTracker.detect()` для зап'ясть (кожен 4-й кадр), застосовує `CameraSelector.select()` для кожного суглоба та `AngleEngine.compute_angles()` для отримання списку `AngleResult`. Агрегований `snapshot (angle_results + rendered_bgr)` атомарно оновлюється для доступу з боку GUI, FastAPI-сервера та WebRTC-стріму. При втраті RTSP-з'єднання (відсутність нового кадру впродовж `RTSP_STALE_SEC = 2,5` с) `CaptureThread` автоматично закриває поточний `VideoCapture`, очікує `RTSP_RECONNECT_SEC = 2,0` с та послідовно перебирає URL-кандидати з

кортежу RTSP\_CANDIDATES, спочатку пробуючи CAP\_FFMPEG backend, що забезпечує відмовостійку роботу в умовах нестабільної мережі [47] (Додаток Б).

Модуль `render.py` відповідає за накладання комплексної візуальної інформації на відеокадри у реальному часі. Для відображення україномовних підписів на OpenCV-кадрах використовується бібліотека Pillow: OpenCV-кадр у форматі BGR конвертується у PIL.Image у форматі RGB, текст рендериться засобами PIL з підтримкою TrueType-шрифтів з кирилицею (функція `_find_font()` шукає шрифт на типових системних шляхах Windows, Linux та macOS), після чого зображення конвертується назад у numpy array для подальшої обробки OpenCV. Метод `draw_skeleton()` малює лінії скелетних зв'язків між суглобовими точками (на основі POSE\_CONNECTIONS для 17 точок COCO) та кольорові маркери для кожної анатомічної точки: зеленим - якщо кут суглоба в межах ROM-норми за AAOS, жовтим - якщо наближується до межі (`warn`), червоним - якщо поза нормою (`bad`), що забезпечує миттєву візуальну індикацію клінічного стану. Метод `draw_angles()` рендерить дугу кожного кута безпосередньо на відеокадрі з числовим значенням у градусах: кутовий діапазон дуги обчислюється через `atan2` від координат трьох точок кутового трикутника, що дозволяє точно відобразити напрямок та величину кута у піксельному просторі [47].

Модуль `main.py` реалізує повнофункціональний графічний інтерфейс на базі PyQt5 із трьома функціональними зонами: заголовок із назвою системи, індикатором FPS та статусом підключення камер; центральна область із відображенням відеопотоків від трьох камер; панель керування сесією з кнопками Start/Stop та відображенням поточних ROM-значень. Мікросервіс підтримує два режими відображення: режим сітки (`grid mode`), що одночасно показує всі три камери в одному вікні у конфігурації 1×3, та режим слайдшоу (`slideshow mode`), що циклічно чергує між камерами з налаштовуваним періодом 3 с - корисний при роботі на моніторах з обмеженою роздільною здатністю. Модальний діалог калібрування (`CalibrationDialog`) відстежує прогрес ініціалізації камер та валідації геометрії розміщення, сигналізує про готовність системи до вимірювань та автоматично закривається при досягненні стану READY для всіх трьох камер.

Панель керування сесією відображає поточні значення 22 кутових параметрів у таблиці з кольоровим кодуванням відхилень від норми та керує записом JSON-логу через SessionRecorder [47].

На рисунку 4.2 наведено інтерфейс процедури калібрування трьох IP-камер мікросервісу відеоаналізу, де статус «Готово 3/3» підтверджує успішне підключення та синхронізацію всіх камер. На рисунку 4.3 показано результат детекції кутів суглобів у реальному часі: нейромережевий детектор YOLOv8-pose ідентифікує 17 анатомічних точок скелету, MediaPipe Hand Landmarker додатково визначає 21 точку кисті для зап'ястних кутів, а алгоритм геометричного злиття на базі argmax-правила обчислює значення 22 кутових параметрів з відображенням числових значень безпосередньо на відеокадрі.

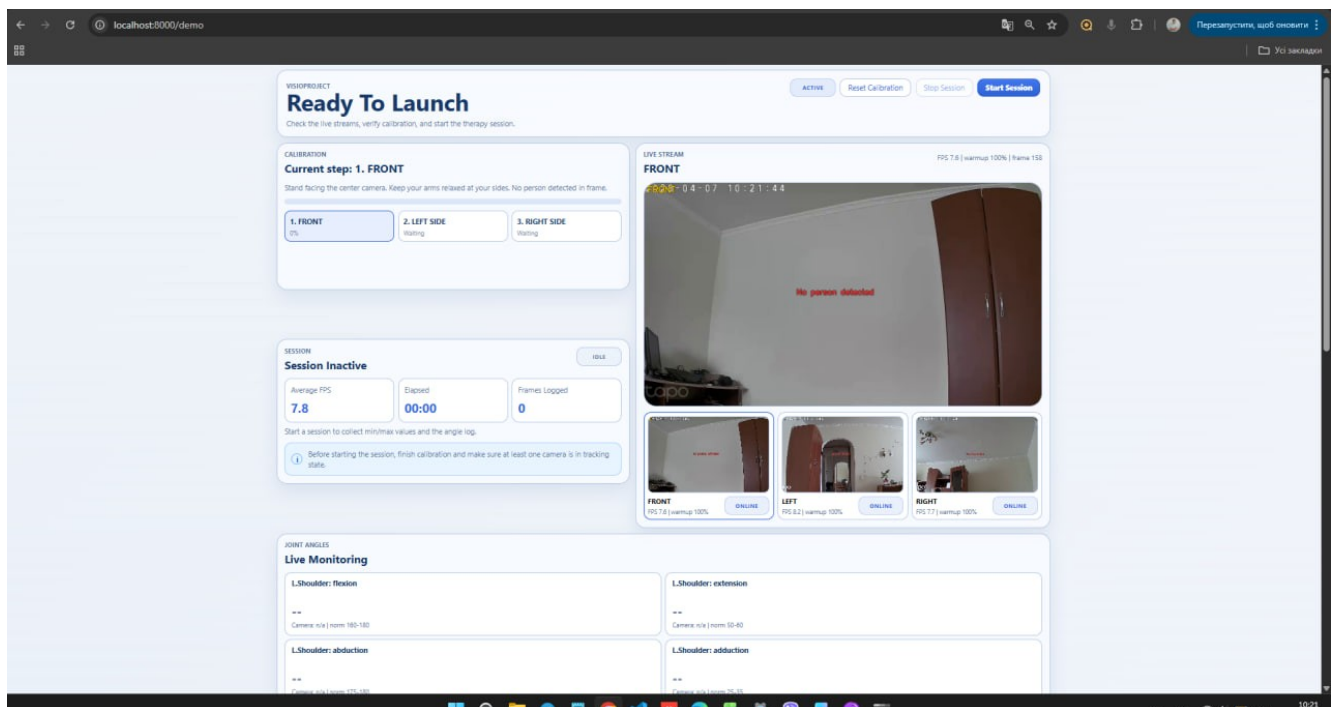


Рис. 4.2. Веб-клієнт автономного гоніометричного моніторингу: панель калібрування та стану камер

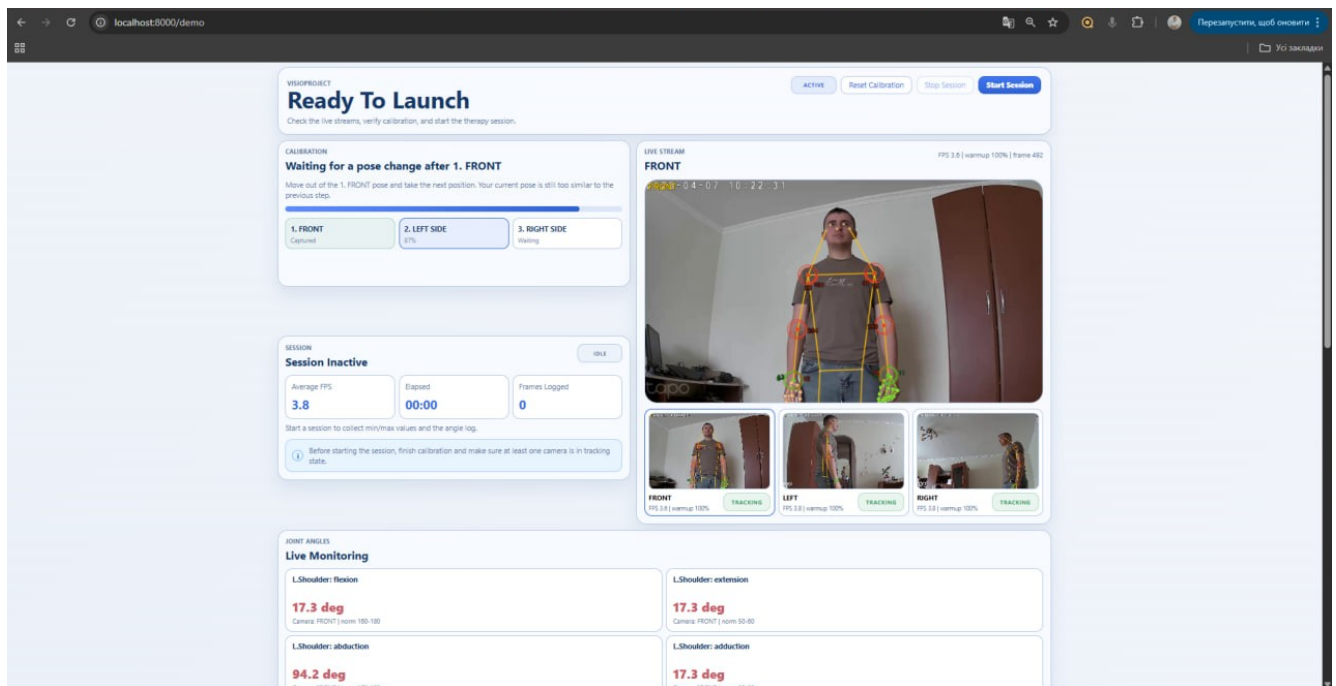


Рис. 4.3. Веб-клієнт автономного гоніометричного моніторингу: відеотрансляція з візуалізацією скелету та панель моніторингу 22 ROM-параметрів

Загальний потік даних у мікросервісі організовано наступним чином. Три CaptureThread безперервно захоплюють кадри від IP-камер через RTSP/TCP [130] та атомарно зберігають результати у `cam_frames`. Центральний `asyncio event loop` у `pipeline.py` з частотою 30 Гц зчитує `cam_frames` від усіх камер, виконує `batched-інференс YOLOv8-rose` та `OneEuro-згладжування`, кожен 4-й кадр додатково викликає `MediaPipe Hand Landmarker`, виконує детермінований вибір камери для кожного суглоба через `CameraSelector` (2.13) та обчислення кутів через `AngleEngine`, і передає агрегований `snapshot` трьом споживачам: GUI-потоків `PyQt5` для візуалізації в інтерфейсі, `FastAPI`-серверу для доступу через REST-API та `WebRTC`-стріму для відеотрансляції клієнтам. `SessionRecorder` паралельно записує кутові дані у JSON-файл сесії з покадровою структурою. По завершенні сесії зведення ROM-параметрів передається через REST API серверній частині (підрозділ 3.4) для збереження у PostgreSQL та подальшого інтервального моделювання [48].

Мережева інфраструктура побудована на роутері у режимі бездротового репітера, що забезпечує незалежну підмережу для IP-камер зі статичними адресами у 192.168.1.0/24. При переносі системи до нової клінічної установи оператор підключає лише хост-комп'ютер до роутера - IP-адреси та RTSP URL камер залишаються незмінними. Вибір протоколу RTSP/TCP замість UDP обумовлений необхідністю надійної доставки в умовах NAT-трансляції роутера та можливих мережевих втрат у Wi-Fi середовищі. FFmpeg-параметри через `OPENCV_FFMPEG_CAPTURE_OPTIONS` (`fflags=nobuffer`, `flags=low_delay`, `max_delay=500000`) зменшують системну затримку відеосигналу з типових 2-4 с до 80-200 мс [47]. Параметри RTSP-з'єднання наведено у таблиці 4.1.

Таблиця 4.1.

Параметри RTSP-з'єднання мікросервісу відеоаналізу

Параметр	Значення	Призначення
RTSP_PORT	554	Стандартний порт RTSP (IANA)
RTSP_USE_TCP	True	Транспорт RTP/TCP замість UDP
RTSP_BUFFER_SIZE	1	Мінімальний буфер VideoCapture (зменшення затримки)
RTSP_OPEN_TIMEOUT_SEC	6,0	Тайм-аут відкриття з'єднання
RTSP_READ_TIMEOUT_SEC	4,0	Тайм-аут читання кадру
RTSP_STALE_SEC	2,5	Поріг часу застарілого кадру (offline-детекція)
RTSP_RECONNECT_SEC	2,0	Інтервал між спробами перепідключення
RTSP_FFMPEG_OPTIONS	<code>nobuffer;</code> <code>low_delay</code>	FFmpeg: мінімальна буферизація та затримка

FFmpeg-параметри передаються через змінну середовища `OPENCV_FFMPEG_CAPTURE_OPTIONS` перед відкриттям VideoCapture.

Параметр `fflags=nobuffer` вимикає буферизацію пакетів у FFmpeg-декодері, `flags=low_delay` вмикає режим мінімальної затримки декодування, `max_delay=500000` обмежує максимальну затримку буфера 0,5 секундами. Сукупний ефект цих налаштувань - зменшення системної затримки відеосигналу від камери до екрану з типових 2-4 с (при буферизації за замовчуванням) до 80-200 мс, що є прийнятним для інтерактивного клінічного використання [47].

Клас `CaptureThread` реалізує відмовостійке з'єднання з RTSP-камерою в окремому `daemon`-поточці. Метод `_open_capture()` послідовно перебирає URL-кандидати (`stream2`, `stream1`), спочатку пробуючи FFmpeg backend (`cv2.CAP_FFMPEG`), а у разі невдачі - автоматичний вибір backend. Метод `is_online()` визначає актуальність з'єднання за часом останнього кадру: якщо від моменту отримання кадру пройшло більше `RTSP_STALE_SEC = 2,5` с, камера вважається `offline` і ініціюється процедура перепідключення. Функція `_mask_rtsp_url()` маскує пароль у логах, замінюючи його на «\*\*\*», що є необхідним заходом безпеки при записі URL до журналів [47].

Точність кутових вимірювань реалізованої системи визначається трьома основними джерелами похибки: похибкою нейромережевого детектора YOLOv8-pose, проекційною похибкою вимірювання з використанням однієї камери та залишковим шумом після OneEuro-фільтрації. Аналіз кожного з цих джерел дозволяє оцінити сумарну точність системи та обґрунтувати її відповідність клінічним вимогам [122, 132].

Детектор YOLOv8-pose має середню похибку позиціонування ключових точок на рівні MediaPipe Pose Landmarker ( $\approx 55-60$  мм на 3D-бенчмарках при меншій кількості точок) - цей показник транслюється у кутову похибку, величина якої залежить від довжини кісткових сегментів суглоба та геометрії конкретного руху. Для типового ліктьового суглоба з довжиною плеча  $\sim 30$  см та передпліччя  $\sim 25$  см кутова похибка становить  $\approx 3-8^\circ$  залежно від поточного кута згинання. Для плечового суглоба, де кісткові сегменти довші, кутова похибка дещо менша -  $\approx 2-6^\circ$ . Ці значення відповідають концепції мінімальної клінічно значимої різниці (MCID), що для ROM зазвичай становить  $5-15^\circ$  залежно від суглоба та типу

патології [72], що свідчить про клінічну придатність системи для моніторингу реабілітаційного прогресу [21].

Проекційна похибка є домінуючим джерелом помилки в безмаркерних системах із однією камерою. Аналітична оцінка, наведена у підрозділі 2.1 (формула 2.1), демонструє, що при куті відхилення оптичної осі камери від площини руху  $\varphi > 45^\circ$  абсолютна похибка вимірювання перевищує  $30^\circ$  для типових кутів суглобів. У таблиці 4.2 наведено результати аналізу проекційної похибки для основних рухів суглобів верхніх кінцівок при конфігурації із трьома камерами [47].

Таблиця 4.2.

## Очікувана проекційна похибка для системи із трьома камерами

Рух суглоба	Площина руху	$\varphi$ (FRONT)	$\varphi$ (L/R)	Обрана камера / похибка
Shoulder flexion	Сагітальна	$0-10^\circ$	$80-90^\circ$	FRONT, $< 2^\circ$
Shoulder abduction	Фронтальна	$\sim 0^\circ$	$\sim 90^\circ$	FRONT, $< 2^\circ$
Elbow flexion L	Сагітальна L	$\sim 80^\circ$	$\sim 5-10^\circ$	LEFT, $< 3^\circ$
Elbow flexion R	Сагітальна R	$80^\circ$	$\sim 5-10^\circ$	RIGHT, $< 3^\circ$
Wrist bend	Сагітальна	$\sim 70^\circ$	$\sim 10^\circ$	L/R side, $< 4^\circ$

З таблиці 4.2 видно, що система із трьома камерами та алгоритмом детермінованого аргмакс-вибору камери (CameraSelector) забезпечує проекційну похибку менше  $5^\circ$  для всіх відстежуваних рухів верхніх кінцівок, тоді як система із однією фронтальною камерою мала б похибку  $30-50^\circ$  при вимірюванні ліктьового та зап'ясткового згинань у сагітальній площині. Таким чином, підхід із трьома камерами зменшує максимальну проекційну похибку у 6-10 разів порівняно з системами на базі однієї камери [122, 132].

Продуктивність реалізованого мікросервісу оцінено на робочій станції із процесором Intel Core i7-10700 (8 ядер, 16 потоків) та 16 ГБ оперативної пам'яті (таблиця 4.3). Домінуючим за часом етапом є нейромережевий інференс YOLOv8-

pose (8-12 мс/кадр з OpenVINO на процесорі), що обмежує частоту обробки однієї камери до 80-125 кадрів за секунду. При застосуванні субсемплінгу  $POSE\_EVERY\_N\_FRAMES = 3$  фактична частота інференсу знижується до  $\approx 25-40$  Гц при цільовій частоті кадрів 30 FPS, що є достатнім для клінічних вимірювань з огляду на повільну динаміку рухів при реабілітації.

Таблиця 4.3.

## Обчислювальна складність та час виконання етапів конвеєру

Етап конвеєру	Складність	Час (мс)	Паралелізм
VideoCapture.read() (RTSP)	-	10-30	По потоку камери
_downscale_for_inference()	$O(W \times H)$	2-5	По потоку камери
InferenceWorker.predict() (YOLOv8 + OpenVINO)	$O(W' \times H')$	8-12	Batched (3 кадри)
KeypointSmoother.apply() $\times$ 34 фільтри	$O(102) \approx O(1)$	< 0,1	По потоку камери
HandTracker.detect() (кожен 4-й кадр)	-	10-15	Окремий потік
AngleEngine.compute_angles() (22 кутів)	$O(22)$	< 0,5	GUI / async
Renderer.draw_*()	$O(W \times H)$	3-8	По потоку камери
CameraSelector.select() + GUI refresh	$O(3 \times 22)$	< 1	GUI-потік

Паралельне виконання трьох CaptureThread-потоків дозволяє одночасно обробляти три камери без збільшення латентності, оскільки кожен потік виконує лише захоплення кадру, а основний обчислювально-вартісний конвеєр централізовано виконується у pipeline.py з batched-інференсом. Загальне

навантаження CPU при трьох потоках обробки становить 35-55%, RAM -  $\approx$  850 МБ. Мережеве навантаження при роздільній здатності 1280×720 та H.264 стиску для трьох потоків становить  $V_{total} = 12$  Мбіт/с; рекомендована пропускна здатність мережевого каналу -  $\geq 20$  Мбіт/с, що забезпечується стандартним Ethernet 100BASE-TX або Wi-Fi 802.11n з п'ятикратним запасом [47].

Домінуючим за часом етапом є нейромережевий інференс YOLOv8-pose ( $\sim$ 8-12 мс/кадр) та MediaPipe Hand Landmarker ( $\sim$ 10-15 мс/кадр, виконується лише 1 раз на 4 кадри). Паралельне виконання захоплення на багатоядерному CPU дозволяє одночасно обробляти три камери, збільшуючи загальну пропускну здатність без збільшення латентності. Загальне навантаження CPU при трьох потоках обробки з субсемплінгом ( $POSE\_EVERY\_N\_FRAMES = 3$ ,  $HANDS\_EVERY\_N\_FRAMES = 4$ ) становить 35-55%, RAM -  $\approx$  850 МБ, що залишає суттєвий запас обчислювальних ресурсів для розширення функціональності (додавання AR-рендерингу для гарнітури Magic Leap 2, паралельного запуску модуля інтервального прогнозування тощо) [47].

Мережеве навантаження при роздільній здатності 1280×720 та стиску H.264 (типовий бітрейт IP-камер 2-4 Мбіт/с) для трьох потоків становить  $V_{total} = 3 \cdot 4 = 12$  Мбіт/с. З урахуванням RTSP/RTP/TCP overhead (+5-15%) та запасу надійності рекомендована пропускна здатність мережевого каналу складає  $\geq 20$  Мбіт/с. Стандартний Ethernet 100BASE-TX (100 Мбіт/с) або Wi-Fi 802.11n (54-300 Мбіт/с) забезпечують цю вимогу з п'ятикратним запасом [47].

Розроблений мікросервіс має ряд обмежень, які визначають межі її застосовності та напрями подальшого вдосконалення. По-перше, система здійснює вимірювання кутів у проекції пікселів кадру, що є 2D-апроксимацією 3D-кінематики суглобів. При непаралельності площини руху та площини зображення залишається остаточною проекційна похибка, мінімізована, але не повністю усунена підходом із трьома камерами. По-друге, точність визначення координат суглобових точок обмежена продуктивністю детектора YOLOv8-pose ( $MPJPE \approx 55-60$  мм), що транслюється у кутову похибку 3-8°. По-третє, три камери не синхронізовані апаратно, що при швидких рухах може спричинити похибку через часове зміщення

кадрів (~33 мс при 30 FPS). По-четверте, YOLOv8-pose визначає лише 17 точок СОСО проти 33 точок у MediaPipe Pose, через що деталізація кистей здійснюється окремим викликом MediaPipe Hand Landmarker, що частково збільшує обчислювальне навантаження та вимагає процедури L/R-вирівнювання виявлених кистей [47].

Перспективними напрямками подальшого розвитку мікросервісу є: стереоскопічна 3D-реконструкція координат суглобових точок через калібрування intrinsic/extrinsic параметрів камер та триангуляцію методом DLT; апаратна часова синхронізація кадрів через PTP (Precision Time Protocol, IEEE 1588); інтеграція з HL7 FHIR для передачі вимірювань ROM у системи електронних медичних записів (EHR); застосування рекурентних нейронних мереж (LSTM, Transformer) для класифікації патологічних рухових патернів на основі накопичених часових рядів кутових вимірювань; можливість перемикання між YOLOv8-pose та MediaPipe Pose у конфігурації для сценаріїв із достатнім обчислювальним ресурсом, коли пріоритетом є максимальна деталізація анатомічних точок [47].

Таким чином, програмна архітектура мікросервісу автоматичної гоніометрії реалізує принципи модульності, розділення відповідальностей та мінімізації зв'язності у вигляді одинадцяти спеціалізованих модулів із чітко визначеними інтерфейсами. Багатопотокова архітектура producer-consumer із daemon-потокami CaptureThread забезпечує паралельне захоплення трьох відеопотоків без взаємного блокування, а централізований asyncio event loop у pipeline.py координує інференс YOLOv8-pose, OneEuro-згладжування координат, детермінований вибір камери та обчислення кутів. Ієрархія класів KeypointSmoother → OneEuroFilter та CameraSelector → AngleEngine реалізує програмний каркас для математичного апарату оптимізованої реалізації методу (підрозділ 2.1). Веб-клієнт автономного моніторингу (vision\_web\_api.py + web\_demo/) розширює функціональність мікросервісу для сценаріїв без AR-гарнітури, забезпечуючи повний доступ до гоніометричних даних, відеопотоків та управління сесіями через браузер без додаткових залежностей. Мікросервіс інтегрується з серверною частиною через REST API, передаючи ROM-зведення для побудови інтервальних моделей

динаміки відновлення (підрозділ 2.2) та формування рекомендацій щодо корекції терапевтичного протоколу (підрозділ 2.3).

#### **4.2. Реалізація AR-застосунку арт-терапії для гарнітури Magic Leap 2**

Графічне нанесення малюнка реалізується модулем TexturePainter, архітектура якого побудована в обхід стандартних засобів лінійного рендерингу (Line Renderer) для зменшення навантаження на обчислювальний модуль системи на кристалі гарнітури Magic Leap 2. Замість генерування нових геометричних примітивів для кожного руху пензля, модуль виконує прямі бітові маніпуляції над масивом пікселів віртуальної текстури. TexturePainter отримує локальні UV-координати двовимірної поверхні, до якої торкається просторовий промінь, та переводить UV-пропорції в цілочисельні індекси масиву текстури. Алгоритм зафарбовує кругову область заданого радіусу у вказаний колір, перевіряючи приналежність кожного пікселя до кола через порівняння квадратів відстаней ( $dx^2 + dy^2 \leq \text{radiusSq}$ ) без обчислення квадратного кореня, що є оптимальним з погляду продуктивності [48].

Критичною проблемою класичних систем модифікації UV-координат є деформація (stretch) області нанесення – перетворення віртуального круглого пензля в еліпсоподібний – у випадках, коли цільове полотно є прямокутним, тобто коли співвідношення ширини до висоти текстури відрізняється від одиниці. У модулі TexturePainter реалізовано автоматичну компенсацію деформації на базі розрахунку співвідношення сторін (Aspect Ratio) зображення-основи. При  $\text{aspectRatio} > 1$  (ширина більша за висоту) масштабний коефіцієнт по вертикальній осі коригується множенням на  $\text{aspectRatio}$ ; при  $\text{aspectRatio} < 1$  (висота більша за ширину) – масштабний коефіцієнт по горизонтальній осі коригується діленням на  $\text{aspectRatio}$ . Така корекція гарантує формування правильного кола незалежно від розмірів та пропорцій терапевтичних зображень-розмальовок, що експортуються на полотно з вебзастосунку фізіотерапевта. Рендеринг через модифікацію буфера текстури залишається високопродуктивним при обробці полотен із роздільною здатністю  $1024 \times 1024$  пікселів, що є достатнім для терапевтичних завдань арт-терапії [48].

Оскільки процес арт-терапії потребує динамічного залучення фізіотерапевта для оцінки когнітивних і моторних функцій пацієнта, застосунок має інтеграційний модуль двонаправленого мережевого зв'язку, побудований за принципом клієнт-серверної архітектури. Модуль ініціалізації середовища зв'язується із REST API серверної частини NestJS (підрозділ 3.4) під час етапу ініціалізації корутини, використовуючи HTTP-запити через стандартний клас `UnityWebRequest`. Сесія пацієнта надсилає параметри ідентифікації пристрою, а у відповідь отримує пакет інструкцій від панелі адміністрування фізіотерапевта: параметри оточення AR-сцени, дизайн та форму полотна, кут нахилу та висоту відносно рівня очей. Зміна будь-якого параметра фізіотерапевтом на екрані конфігурації AR-сцени вебзастосунку негайно транслюється на гарнітуру через `WebSocket` push-повідомлення від серверної частини NestJS без необхідності перезапуску AR-застосунку, що є ключовою клінічною функцією адаптивного управління реабілітаційним процесом [48].

Окрім конфігураційних параметрів, через REST API динамічно завантажуються терапевтичні зображення-розмальовки, які розміщуються на віртуальному полотні безпосередньо під час перебування пацієнта у AR-середовищі. Це дозволяє фізіотерапевту проводити гнучку терапію, поступово підвищуючи або зменшуючи моторну складність елементів малюнка без переривання сеансу. Зображення генеруються рекомендаційною системою підтримки (підрозділ 3.4) на основі прогнозованих інтервальних значень кутів згину суглобів та автоматично завантажуються на AR-полотно. Разом із цим, для забезпечення спостереження фахівця у повноцінному наскрізному режимі (`First-Person View`), інтеграційний модуль реалізує трансляцію відеопотоку AR-середовища за протоколом `WebRTC`. Відеопотік формується з вбудованої RGB-камери гарнітури `Magic Leap 2` з роздільною здатністю  $1920 \times 1080$  пікселів, частотою 30 кадрів на секунду та кодеком `H.264 Baseline`. Відеопотік комбінується з AR-оверлеями – візуалізацією скелету руки, контурами розмальовок та елементами інтерфейсу – та подається на віддалену станцію фізіотерапевта із затримкою близько 60 мілісекунд в умовах локальної мережі реабілітаційного

кабінету. Сигнальна частина WebRTC-з'єднання (обмін SDP-дескрипторами та ICE-кандидатами) делегована серверній частині NestJS через модуль WebRtcModule (підрозділ 3.4); після встановлення P2P-з'єднання медіадані передаються безпосередньо між гарнітурою та браузером фізіотерапевта, минаючи сервер [48].



Рис. 4.4. Інтерфейс AR-застосунку арт-терапії на гарнітурі Magic Leap 2

На рисунку 4.4 наведено знімок екрану AR-інтерфейсу арт-терапії, виконаного на гарнітурі Magic Leap 2. Інтерфейс складається з трьох візуальних компонентів: центральне полотно з терапевтичним зображенням-розмальовкою, панель кольорів для вибору відтінку пензля та кнопка-ластик для видалення фрагментів малюнка. Полотно автоматично позиціонується модулем WorkspaceInitializer на відстані 0,75 м від пацієнта та орієнтується обличчям до нього. Пацієнт взаємодіє з полотном через жести відстеження рук (Hand Tracking) без використання фізичних контролерів, що є принциповою перевагою для

реабілітації верхніх кінцівок, оскільки природні рухи руки безпосередньо відповідають цільовим рухам терапевтичних вправ [41, 50].

Загальний потік даних AR-застосунку організований через три комунікаційних канали із серверною частиною (підрозділ 3.4). Канал конфігурації: при ініціалізації сесії застосунок отримує від REST API серверної частини повний набір параметрів AR-середовища (оточення, дизайн полотна, геометрію) та URL терапевтичного зображення; під час сесії оновлення параметрів надходять через WebSocket push. Канал медіаконтенту: терапевтичні зображення-розмальовки асинхронно завантажуються за URL, отриманим від сервера, та розміщуються на текстурі полотна без перезавантаження сцени. Канал відеотрансляції: WebRTC P2P-з'єднання передає комбінований відеопотік (реальна кімната + AR-оверлеї) безпосередньо на браузер фізіотерапевта із затримкою близько 60 мс. Такий дистанційний моніторинг у поєднанні з обробленою серверною частиною метрикою рухів руки гарантує своєчасну візуально-математичну валідацію лікувальних заходів та можливість миттєво коригувати параметри арт-терапії залежно від стану рухового апарату пацієнта [48].

Таким чином, програмна архітектура AR-застосунку арт-терапії для гарнітури Magic Leap 2 побудована на модульному принципі з трьома ключовими підсистемами. Модуль `WorkspaceInitializer` забезпечує автоматичне просторове позиціонування робочої зони на основі вектора погляду пацієнта з корекцією зміщення опорної точки, що гарантує ергономічну позу руки у межах терапевтичного діапазону рухів. Модуль `BrushManager` реалізує метод ретроспективного трасування променя із зсувом 0,05 м для подолання ефекту проколювання голограм та диференціює взаємодію з інтерфейсними елементами та полотном через механізм бітових масок шарів. Модуль `TexturePainter` виконує прямі бітові маніпуляції над масивом пікселів текстури з автоматичною компенсацією деформації пензля через коефіцієнт `aspectRatio`, забезпечуючи продуктивний рендеринг при роздільній здатності  $1024 \times 1024$  пікселів. Інтеграційний модуль мережевої комунікації забезпечує динамічне завантаження конфігурації сесії та терапевтичних зображень через REST API серверної частини,

push-оновлення параметрів через WebSocket та WebRTC-трансляцію комбінованого відеопотоку з AR-оверлеями для дистанційного клінічного спостереження фізіотерапевтом [48].

#### **4.3. Реалізація центрального сервера-оркестратора з мікросервісом рекомендацій та вебзастосунку фізіотерапевта**

У функції bootstrap серверного застосунку реєструються три глобальних провайдери, що застосовуються до всього API як крос-функціональні компоненти. ValidationPipe перехоплює всі вхідні запити перед тим, як вони досягнуть контролера: параметри whitelist та forbidNonWhitelisted виключають передачу незадекларованих полів, а transform із enableImplicitConversion забезпечує автоматичне перетворення типів примітивів. HttpExceptionHandler є централізованим обробником виключень, що перехоплює як NestJS HttpException, так і нативні JavaScript Error, формуючи стандартизовану JSON-відповідь зі структурою {success, statusCode, timestamp, path, error}. ResponseInterceptor реалізує патерн Interceptor та обгортає будь-яку успішну відповідь контролера у стандартну структуру {success: true, statusCode, data, timestamp}. Комбінація цих трьох провайдерів уніфікує формат усіх відповідей API, спрощує розробку клієнтського коду та забезпечує консистентну діагностику помилок [48].

Модуль WebRtcModule забезпечує двосторонню аудіо- та відеокommunікацію між веббраузером фізіотерапевта та гарнітурою Magic Leap 2 пацієнта. Комунікаційний канал базується на стандарті WebRTC (Web Real-Time Communication), специфікованому W3C та IETF (RFC 8825-8835), що забезпечує peer-to-peer передачу медіапотоків із наскрізним шифруванням DTLS-SRTP без необхідності центрального медіасервера. Архітектурно модуль складається з двох компонентів: WebRtcGateway (сигнальний шлюз) та RoomService (управління кімнатами). Сигнальний шлюз реалізований як NestJS WebSocket Gateway з використанням бібліотеки Socket.IO та виступає посередником для обміну SDP (Session Description Protocol) офферами та відповідями, а також ICE (Interactive Connectivity Establishment) кандидатами між пірами. Протокол встановлення з'єднання передбачає шість кроків: створення кімнати (POST /api/v1/rooms),

під'єднання гарнітури через WebSocket (join-room), надсилання SDP offer від терапевта, відповідь SDP answer від гарнітури, обмін ICE-кандидатами для NAT Traversal та встановлення прямого P2P DTLS-SRTP каналу [48].

Клас RoomService відповідає за життєвий цикл сеансів комунікації. Кожна кімната ідентифікується унікальним UUID v4 та асоціюється із записом сесії, що зберігається у пам'яті сервера (in-memory Map) протягом часу сеансу. Сервіс відстежує стан кімнати через цикл WAITING → ACTIVE → CLOSED: при створенні кімната отримує статус WAITING, при під'єднанні пацієнта – ACTIVE, а після від'єднання обох учасників автоматично закривається із звільненням ресурсів. Після завершення ICE-узгодження медіадані передаються безпосередньо між пристроями, минаючи сервер, що мінімізує навантаження на серверну інфраструктуру та забезпечує затримку відеопотоку близько 60 мс в умовах локальної мережі реабілітаційного кабінету [48].

Модуль інтервального прогнозування (Interval Forecasting Module) інтегрований у серверну частину через C++-адаптер, що інкапсулює виклики до нативної DLL інтервальної ідентифікації математичних моделей, та реалізує програмний інтерфейс для побудови та використання інтервальних математичних моделей динаміки відновлення кутів рухливості суглобів, математичний апарат яких описано у підрозділі 2.2. Модуль отримує з бази даних PostgreSQL накопичені часові ряди кутових вимірювань за проведеними сеансами конкретного пацієнта та, у разі втрати адекватності поточної моделі (підрозділ 2.3), делегує бібліотеці виконання структурної та параметричної ідентифікації нової інтервальної моделі на основі алгоритму штучної бджолиної колонії. Результатом роботи модуля є прогнозні інтервали для кожного з контрольованих кутів суглобів на наступні сеанси: для кута згинання (160–180°), розгинання (50–60°) та відведення (180°) плечового суглоба модуль формує нижню та верхню межі очікуваних значень, а також оцінку кількості сеансів до досягнення нормативних порогів. Ці прогнозні дані передаються через REST API на вебзастосунок фізіотерапевта для візуалізації та прийняття клінічних рішень [48].

Екран «Карта пацієнта» (PatientOverview) (рисунок 4.5) є головним екраном застосунку та відображається за замовчуванням. Він надає фізіотерапевту зведений вигляд клінічних показників активного пацієнта та складається з чотирьох функціональних блоків. Телеметрична сітка (TelemetryGrid) відображає ROM-показники (Range of Motion) для правої та лівої сторін у форматі карток із числовим значенням у градусах та дельтою відносно попереднього сеансу. Кінематичний граф прогресу – власна SVG-реалізація лінійного графіка без зовнішніх бібліотек – відображає шість серій даних (плече, лікоть, зап'ясток для кожної сторони) за проведеними вимірювальними циклами з колірним кодуванням відхилень від норми. Прогнозні інтервали, сформовані модулем інтервального прогнозування серверної частини, відображаються на графіку пунктирними смугами, що дозволяє фізіотерапевту візуально порівняти фактичну динаміку з прогнозованою та виявити відхилення від очікуваної траєкторії відновлення. Журнал виконаних сесій та тижневий план реабілітації завершують екран [48].

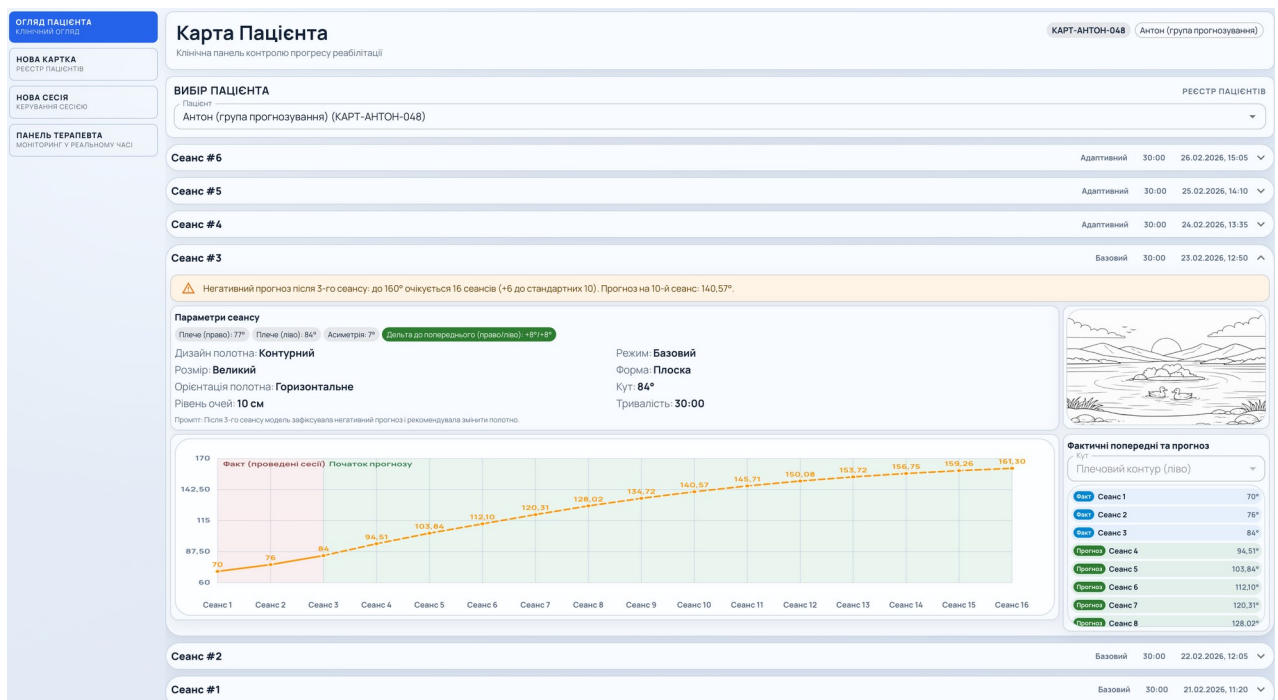


Рис. 4.5. Вебзастосунок фізіотерапевта: екран «Карта пацієнта» з кінематичним графом прогресу та прогнозними інтервалами

Екран «Введення даних пацієнта» (PatientRegistration) забезпечує структуроване введення анамнезу нового пацієнта. Форма складається з трьох

функціональних панелей: ідентифікаційний блок (ПІБ, дата народження, номер медичної картки, код протоколу реабілітації) та операційні нотатки (базовий стан пацієнта, план терапії). Введені дані формують контекст для генерації терапевтичних зображень та визначення початкових нормативних порогів для модуля інтервального прогнозування [48].

Рекомендаційна система підтримки фізичної реабілітації інтегрована безпосередньо у вебзастосунок фізіотерапевта як внутрішній логічний рівень, що розташовується між рівнем відображення даних (React-компоненти екранів) та рівнем серверного API (NestJS) [49]. На відміну від традиційних rule-based рекомендаційних систем у реабілітації, що ґрунтуються на поточних вимірюваннях або статичних клінічних правилах, запропоноване рішення використовує прогнозовані інтервальні значення кутів згину суглобів як основне інформаційне джерело для формування рекомендацій, що дозволяє здійснювати проактивну корекцію реабілітаційного процесу на основі очікуваної динаміки відновлення. Вхідними даними для рекомендаційного модуля є результати роботи модуля інтервального прогнозування серверної частини: прогнозовані інтервальні значення кутів згину суглобів, обчислені за допомогою інтервальних математичних моделей (підрозділ 2.2) на основі часових рядів попередніх реабілітаційних сесій конкретного пацієнта. Структурна ідентифікація визначила, що для адекватної моделі достатньо  $d = 3$  попередніх значень часового ряду, тобто рекомендаційний модуль починає формувати прогнозні траєкторії після третього сеансу [49].

Центральним елементом архітектури рекомендаційної системи є компонент аналізу прогнозу, інтегрований у екран «Карта пацієнта» вебзастосунку. Він порівнює прогнозовані значення кутів з очікуваними або цільовими траєкторіями відновлення, що формуються на основі клінічних стандартів AAOS (згинання  $160-180^\circ$ , розгинання  $50-60^\circ$ , відведення  $180^\circ$  [124]) та індивідуально визначених цілей реабілітації. На цьому етапі обчислюються формалізовані критерії відхилення прогнозованої динаміки від очікуваних траєкторій: оцінюється кількість сеансів до досягнення нормативного діапазону, відхилення від стандартного курсу та прогнозоване значення на 10-й сеанс. У разі виявлення негативного прогнозу

(уповільнення прогресу, плато, ризик перевантаження) система відображає попередження з конкретними числовими показниками на екрані «Карта пацієнта» [49].

Рекомендації формуються у двох напрямках: корекція терапевтичного зображення та налаштування параметрів AR-сесії. Ключовою відмінністю запропонованої рекомендаційної системи від існуючих аналогів є механізм реалізації рекомендацій через корекцію терапевтичного зображення-розмальовки, яке пацієнт малює на віртуальному AR-полотні під час сеансу арт-терапії (підрозділ 3.3). Рекомендації реалізуються не у вигляді прямих інструкцій пацієнту, а шляхом зміни просторового розподілу елементів зображення, що непомітно для пацієнта спрямовує його рухову активність у необхідному напрямку. Просторова структура зображення безпосередньо визначає біомеханічний профіль рухів: вертикально орієнтовані елементи (високі дерева, вежі, вертикальні абстракції) стимулюють згинання плечового суглоба з піднесенням руки угору; горизонтально орієнтовані елементи (пейзажні панорами, широкі мазки) стимулюють відведення та приведення у горизонтальній площині; дрібні деталі стимулюють точні рухи зап'ястка та пальців. Додатковим напрямком є рекомендації щодо параметрів конфігурації AR-сцени: збільшення розміру полотна та його вертикальна орієнтація стимулюють рухи з більшою амплітудою згинання, тоді як горизонтальне розташування – рухи відведення [49].

Для автоматизації переходу від рекомендації до арт-терапевтичного завдання у вебзастосунку реалізовано багатоступеневий AI-контур генерації терапевтичних зображень [49]. Рекомендаційний модуль формує семантичний опис бажаних змін у біомеханічному профілі рухів пацієнта, який передається модулю формування запитів (prompt generation). Даний модуль використовує API великої мовної моделі (LLM) для автоматичної трансформації клінічної рекомендації у структурований текстовий prompt, що описує візуальні характеристики зображення-розмальовки з урахуванням терапевтичних вимог: орієнтацію елементів, їх розподіл у просторі, рівень деталізації та стиль контурів. Отриманий prompt передається до модуля генерації зображень, який через API генеративної моделі створює контурне

зображення, придатне для використання як терапевтична розмальовка на AR-полотні. Згенероване зображення автоматично завантажується через вебзастосунок на AR-полотно гарнітури Magic Leap 2 без перезапуску AR-застосунку – зміна налаштувань негайно відображається у гарнітурі через WebSocket push від серверної частини NestJS. Такий ланцюжок AI-взаємодії дозволяє уникнути жорстко заданих шаблонів арт-завдань та забезпечує адаптацію візуального контенту до індивідуальних особливостей реабілітації пацієнта [49].

Рекомендації відображаються на двох екранах вебзастосунку. На екрані «Карта пацієнта» (PatientOverview) інтерактивний граф прогресу відображає фактичні значення кутів за проведеними сеансами та прогнозні траєкторії пунктирними лініями на наступні сеанси; у разі негативного прогнозу система виводить попередження з конкретними числовими показниками. На екрані «Конфігурація AR-сцени» (SessionConsole) (рисунок 4.6) перед стартом нового сеансу відображається банер рекомендації з описом необхідних змін та кнопкою «Застосувати корекцію», після натискання якої система автоматично адаптує параметри конфігурації сцени та формує клінічний промпт для генерації зображення. Забезпечення пояснюваності рекомендацій реалізовано через надання фізичному терапевту інформації про причини їх формування: прогнозовані відхилення, тенденції зміни кутів та рівень невизначеності – безпосередньо в інтерфейсі вебзастосунку. Таким чином, рекомендаційна система реалізує замкнений контур керування реабілітаційним процесом «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів», описаний у підрозділі 2.3 [49].

Загальний потік даних між серверною частиною та вебзастосунком організований через REST API та WebSocket-канали (рис. 3.1). Канал гоніометричних даних: Python-мікросервіс відеоаналізу (підрозділ 3.2) передає кутові вимірювання через POST /api/v1/sessions/:id/angles у форматі JSON з частотою ~30 Гц; кожен пакет містить мітку часу, ідентифікатор сесії та масив із 22 кутовими значеннями з показником впевненості.

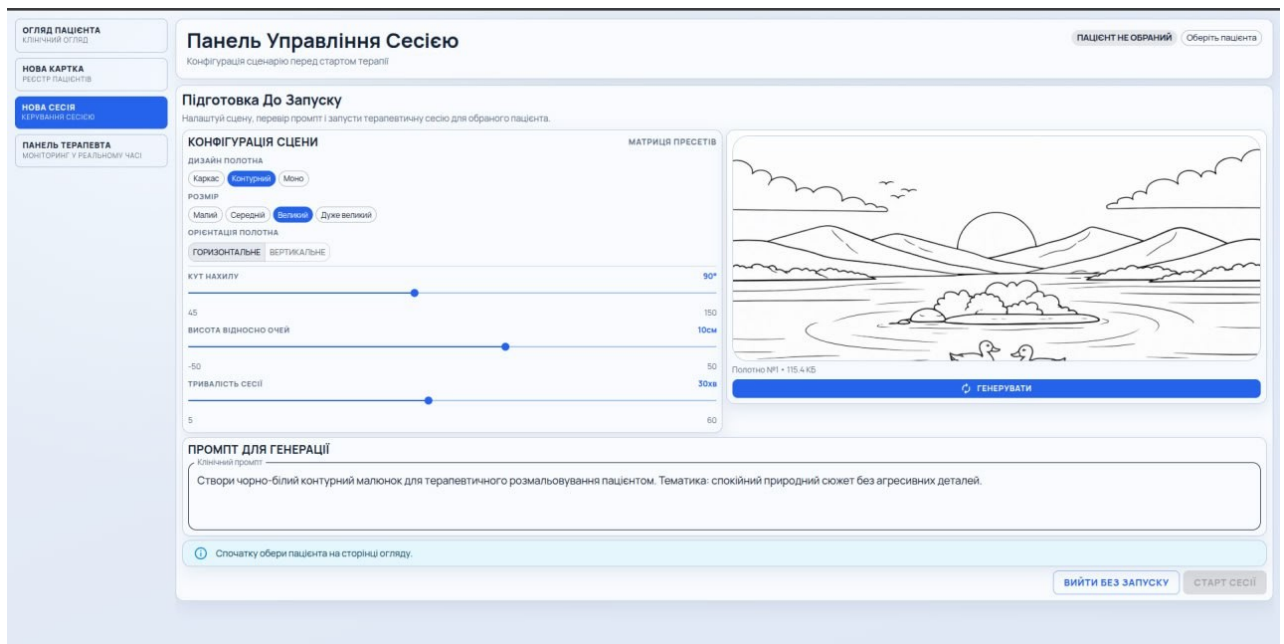


Рис. 4.6. Вебзастосунок фізіотерапевта: екран конфігурації AR-сесії

Серверна частина зберігає ці дані у базі даних PostgreSQL (рис. 3.9) та в реальному часі передає агреговані метрики на вебзастосунок. Канал управління AR-сесією: при зміні параметрів на екрані SessionConsole вебзастосунок надсилає PATCH `/api/v1/sessions/:id` запит до серверної частини, яка негайно транслює оновлення до гарнітури Magic Leap 2 через WebSocket push-повідомлення. Канал прогнозування: після завершення сесії серверна частина автоматично запускає модуль інтервального прогнозування, що обробляє накопичені часові ряди та формує прогнозні інтервали, які передаються на вебзастосунок через GET `/api/v1/patients/:id/forecast` [48].

Вебзастосунок використовує власну дизайн-систему на основі CSS Custom Properties, що визначає набір токенів для кольорів (`--bg-page`, `--bg-panel`, `--accent`, `--ok`, `--warn`), типографіки (шрифт IBM Plex Sans з fallback на Segoe UI) та компонентних стилів. Адаптивна розмітка підтримує три контрольні точки: при ширині менше 1280 пікселів план реабілітації переходить в однорядкове відображення, при ширині менше 1024 пікселів загальна розмітка переходить з двоколонкової у однорядкову, а при ширині менше 760 пікселів усі елементи адаптуються під мобільний пристрій. Збірка проекту виконується через Vite 7 із

плагіном `@vitejs/plugin-react-swc`, що забезпечує компіляцію TypeScript/JSX через SWC-компілятор на мові Rust [48].

Таким чином, програмна архітектура серверної частини та вебзастосунок фізіотерапевта з рекомендаційною системою підтримки побудована на принципах модульності, шаруватої архітектури та розділення відповідальностей. Серверна частина NestJS із шістьма функціональними модулями реалізує REST API для управління пацієнтами, сесіями та медіаконтентом, WebRTC-сигналізацію для відеокommунікації з гарнітурою Magic Leap 2, інтеграцію з модулем інтервального прогнозування та персистентне зберігання даних у реляційній СУБД PostgreSQL. Три глобальних провайдери (`ValidationPipe`, `HttpExceptionFilter`, `ResponseInterceptor`) уніфікують валідацію, обробку помилок та формат відповідей по всьому API. Вебзастосунок React із трьома екранами забезпечує повний робочий цикл фізіотерапевта: перегляд кінематичної аналітики з прогнозними інтервалами, введення даних пацієнта та налаштування параметрів AR-сесії в реальному часі. Рекомендаційна система, інтегрована у вебзастосунок, використовує прогнозовані інтервальні значення кутів згину суглобів для формування рекомендацій щодо корекції терапевтичного зображення-розмальовки та параметрів AR-середовища, реалізуючи механізм непомітної для пацієнта зміни біомеханічного профілю рухів через просторову структуру арт-контенту [49]. Багатоступеневий AI-контур (рекомендація → LLM → prompt → генеративна модель → зображення на AR-полотні) замикає адаптивний цикл «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів», описаний у підрозділі 2.3.

#### **4.4. Інтервальні моделі динаміки відновлення кутів плечового суглоба**

Для верифікації розроблених програмних модулів та побудови інтервальних математичних моделей динаміки відновлення рухливості суглобів проведено клінічне дослідження на базі реабілітаційного відділення КНП «Тернопільська обласна клінічна лікарня» ТОР. Дослідження було організовано як двоетапний послідовний експеримент, що відтворює реалістичний клінічний сценарій впровадження системи: спочатку накопичення даних та побудова прогновної моделі (навчальний етап), а потім – використання побудованої моделі для

активного супроводу нових пацієнтів із механізмом адаптивної корекції терапевтичного протоколу (етап прогнозування, підрозділ 4.5). Усі мікросервіси та застосунки програмного комплексу, архітектуру яких описано у розділі 3, функціонували у повному складі: мікросервіс аналізу трьох відеопотоків з пріоритизацією (підрозділ 3.2), мікросервіс інтервального прогнозування та мікросервіс рекомендацій, оркестровані центральним сервером на NestJS із базою даних PostgreSQL (підрозділ 3.4), AR-застосунок арт-терапії на гарнітурі Magic Leap 2 (підрозділ 3.3) та вебзастосунок фізіотерапевта (підрозділ 3.4).

У дослідженні взяли участь п'ять пацієнтів із різними ортопедичними діагнозами, які були розподілені на дві групи відповідно до етапу дослідження. Навчальна група (три пацієнти) пройшла повний курс із 10 сеансів AR-реабілітації, під час яких здійснювався збір кінематичних даних усіх кутів, які система здатна вимірювати, без використання модуля прогнозування. Зібрані дані навчальної групи використано для побудови інтервальних математичних моделей динаміки відновлення трьох ключових кутів плечового суглоба. Група прогнозування (два пацієнти) проходила реабілітацію на другому етапі дослідження з активним використанням побудованих інтервальних моделей, що дозволяло в реальному часі прогнозувати траєкторію відновлення та формувати рекомендації щодо корекції терапевтичного протоколу; результати апробації описано у підрозділі 4.5.

Як основні показники для побудови інтервальних математичних моделей обрано три ключових кути плечового суглоба, які є критичними для оцінки прогресу реабілітації верхніх кінцівок: кут згинання (flexion) з нормативним діапазоном 160-180°, кут розгинання (extension) з нормативним діапазоном 50-60° та кут відведення (абдукція, abduction) з нормативним значенням 180° [124]. Вибір цих кутів обумовлений кількома факторами: по-перше, згинання, розгинання та відведення є функціонально найбільш значущими рухами плечового суглоба, які безпосередньо задіюються у повсякденній діяльності та виступають ключовими індикаторами відновлення функціональної спроможності; по-друге, під час виконання арт-терапевтичних завдань на AR-полотні ці три рухи домінують у біомеханічному профілі; по-третє, порушення амплітуди цих рухів є спільною

клінічною ознакою для всіх п'яти діагнозів пацієнтів дослідження, що дозволяє застосувати моделі для порівняльного аналізу динаміки відновлення між пацієнтами [48].

Навчальна група складалася з трьох пацієнтів із різним ступенем обмеження рухливості плечового суглоба. Анастасія (25 років, жінка) – субакроміальний бурсит правої верхньої кінцівки; початкове значення кута згинання становило  $140^\circ$ , розгинання –  $40^\circ$ , відведення –  $70^\circ$ , що відповідає помірному обмеженню рухливості. Бурсит характеризується болісністю при активному русі у крайніх положеннях, що зумовлює обережний характер рухової поведінки пацієнта та поступову динаміку відновлення. Валентина (22 роки, жінка) – епікондиліт правого ліктьового суглоба; початкове значення кута згинання плечового суглоба –  $130^\circ$ , розгинання –  $30^\circ$ , відведення –  $160^\circ$ . Хоча первинний діагноз стосується ліктьового суглоба, запальний процес та больовий синдром обмежують загальну рухову активність верхньої кінцівки, включаючи компенсаторне обмеження амплітуди рухів у плечовому суглобі; молодий вік пацієнтки та відсутність структурних ушкоджень суглобової капсули передбачають потенційно високий темп відновлення. Сергій (42 роки, чоловік) – посттравматичний адгезивний капсуліт правого плечового суглоба («заморожене плече»); початкове значення кута згинання –  $60^\circ$ , розгинання –  $25^\circ$ , відведення –  $45^\circ$ , що відповідає тяжкому обмеженню рухливості з дефіцитом  $100-135^\circ$  за різними кутами. Адгезивний капсуліт є найскладнішим із представлених діагнозів, оскільки включає фіброзні зміни суглобової капсули та характеризується тривалим періодом відновлення з нелінійною динамікою [48].

Кожен пацієнт навчальної групи пройшов 10 сеансів AR-реабілітації тривалістю 30 хвилин кожен із періодичністю 2-3 рази на тиждень. Під час кожного сеансу пацієнт надягав гарнітуру Magic Leap 2 і виконував арт-терапевтичне завдання – розмальовування терапевтичного зображення на віртуальному полотні жестама руки в AR-просторі. Початкове зображення-розмальовку було підібрано фізіотерапевтом через екран конфігурації AR-сцени (SessionConsole) вебзастосунку з урахуванням діагнозу та початкового рівня рухливості: для

навчальної групи використовувалося зображення зі збалансованим розподілом елементів по площині полотна, що стимулювало рухи у різних напрямках без акцентування окремих площин руху [48].

Паралельно з виконанням арт-терапевтичного завдання Python-мікросервіс автоматичної гоніометрії (підрозділ 3.2) у реальному часі фіксувала кінематичні дані пацієнтів. Три стаціонарні IP-камери (FRONT, LEFT, RIGHT), розташовані у взаємно перпендикулярних ракурсах відносно робочої зони, передавали відеопотоки по протоколу RTSP/TCP. Нейромережеві детектори YOLOv8-pose та MediaPipe Hands Landmarker ідентифікують 17 та 21 анатомічних точки скелету відповідно, після чого адаптивний фільтр OneEuro (підрозділ 2.1) забезпечував придушення шуму детекції. Модуль геометричного злиття обирає для кожного кутового параметра камеру з найкращим ракурсом на основі показника *confidence visibility*, що зменшувало проєкційну похибку до 3-5°. Система вимірювала всі 22 кутові параметри (11 на кожную сторону тіла) із частотою ~30 кадрів/с, що становило ~54 000 записів за один 30-хвилинний сеанс. Окрім кутових даних, зберігалися службові метадані: версія програмного забезпечення, параметри адаптивного фільтра, якість детекції по кадрах – для забезпечення відтворюваності умов вимірювання між сеансами [48].

Таким чином, за 10 сеансів для кожного пацієнта навчальної групи було сформовано повні часові ряди вимірювань усіх кутових параметрів суглобів верхніх кінцівок. Для побудови інтервальних математичних моделей з кожного часового ряду було отримано 10 агрегованих значень (по одному медіанному значенню на сеанс) та відповідних інтервальних оцінок з урахуванням мінливості вимірювань протягом сеансу. Інтервальні оцінки формувалися на основі квартильного розмаху (IQR) значень кожного кута протягом 30-хвилинного сеансу, що дозволяло коректно врахувати природну варіативність рухів пацієнта під час виконання арт-терапевтичного завдання. Ці інтервальні дані стали вхідними для процедури структурної та параметричної ідентифікації математичних моделей.

На основі зібраних часових рядів кутових вимірювань трьох пацієнтів навчальної групи проведено структурну та параметричну ідентифікацію

інтервальних математичних моделей динаміки відновлення трьох кутів плечового суглоба з використанням алгоритму штучної бджолоїної колонії, математичний апарат якого описано у підрозділі 2.2. Процедура ідентифікації для кожного кута складалася з двох послідовних етапів: структурної ідентифікації, що визначає оптимальну структуру різницевого рівняння (порядок  $d$ , набір базисних функцій), та параметричної ідентифікації, що знаходить оптимальні оцінки вектора параметрів для знайденої структури.

На етапі структурної ідентифікації алгоритм штучної бджолоїної колонії здійснював пошук серед множини моделей-претендентів, мінімізуючи функцію мети (підрозділ 2.2), яка оцінює ступінь узгодженості прогнозованих інтервалів з експериментальними. У результаті структурної ідентифікації було визначено дві оптимальних структури моделей. Для кута згинання ( $160-180^\circ$ ) адекватна модель потребує  $d = 3$  попередніх значень часового ряду як початкових умов для різницевої схеми, тобто для прогнозування значення кута на наступному сеансі достатньо знати результати трьох попередніх сеансів. Для кутів розгинання ( $50-60^\circ$ ) та відведення ( $180^\circ$ ) оптимальна структура моделі потребує  $d = 4$  попередніх значень, що пояснюється більш складною нелінійною динамікою цих рухів та їхньою вищою чутливістю до індивідуальних особливостей патології. Для всіх трьох кутів похибку прогнозування встановлено на рівні  $dZ = 0,1$  – значення, що визначає ширину початкових інтервалів невизначеності. Інтерфейс модуля для задачі структурної ідентифікації наведено на рисунку 4.7 та рисунку 4.8 [48].

На етапі параметричної ідентифікації для кожної знайденої оптимальної структури моделі проводився пошук оцінок вектора параметрів, що забезпечують найкраще узгодження прогнозованих інтервальних значень з експериментальними інтервальними даними трьох пацієнтів навчальної групи. Алгоритм штучної бджолоїної колонії розв'язував оптимізаційну задачу з функцією мети, яка враховує як перетин прогнозованих та експериментальних інтервалів, так і відстань між їхніми центрами у випадку неперетину (підрозділ 2.2). Параметрична ідентифікація виконувалася ітеративно до досягнення умов збіжності. Інтерфейс модуля параметричної ідентифікації наведено на рисунку 4.9 та рисунку 4.10 [48].

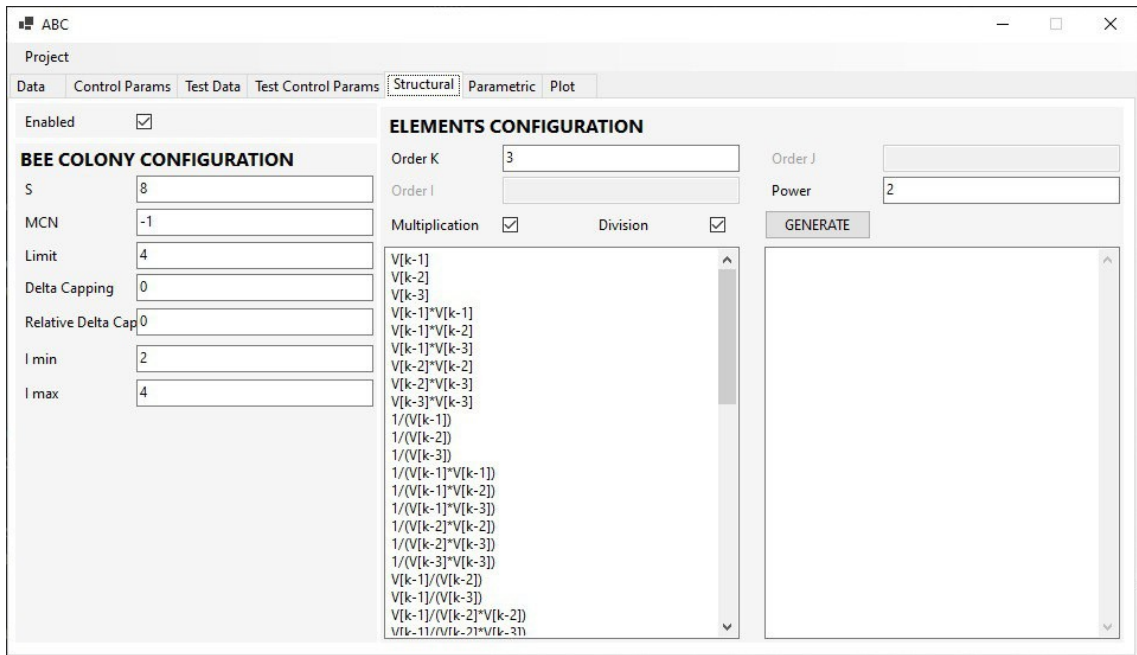


Рис. 4.7. Введення параметрів для структурної ідентифікації інтервальної математичної моделі кута згинання (160-180°)

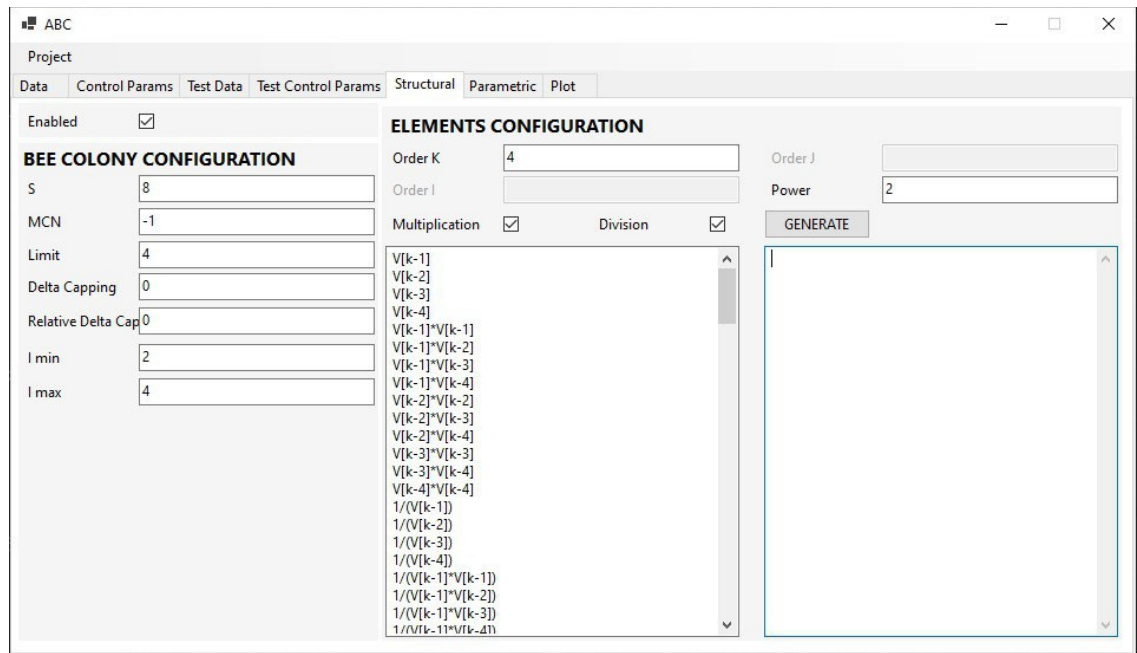


Рис. 4.8. Введення параметрів для структурної ідентифікації інтервальної математичної моделі кутів розгинання (50-60°) та відведення (180°)

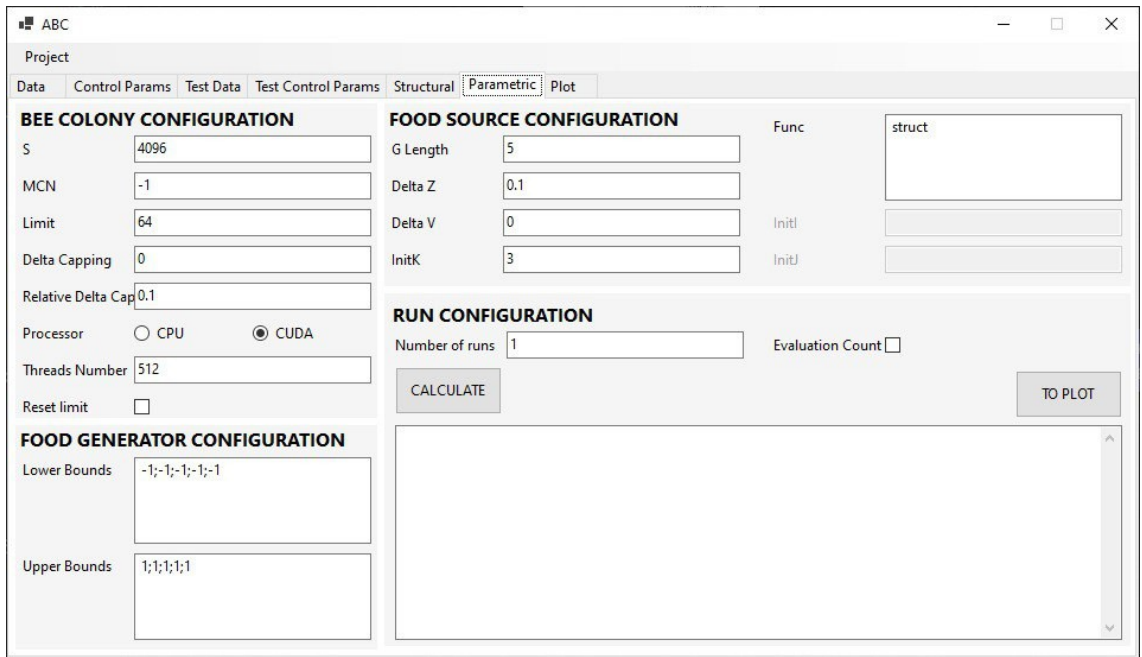


Рис. 4.9. Введення параметрів для параметричної ідентифікації інтервальної математичної моделі кута згинання (160-180°)

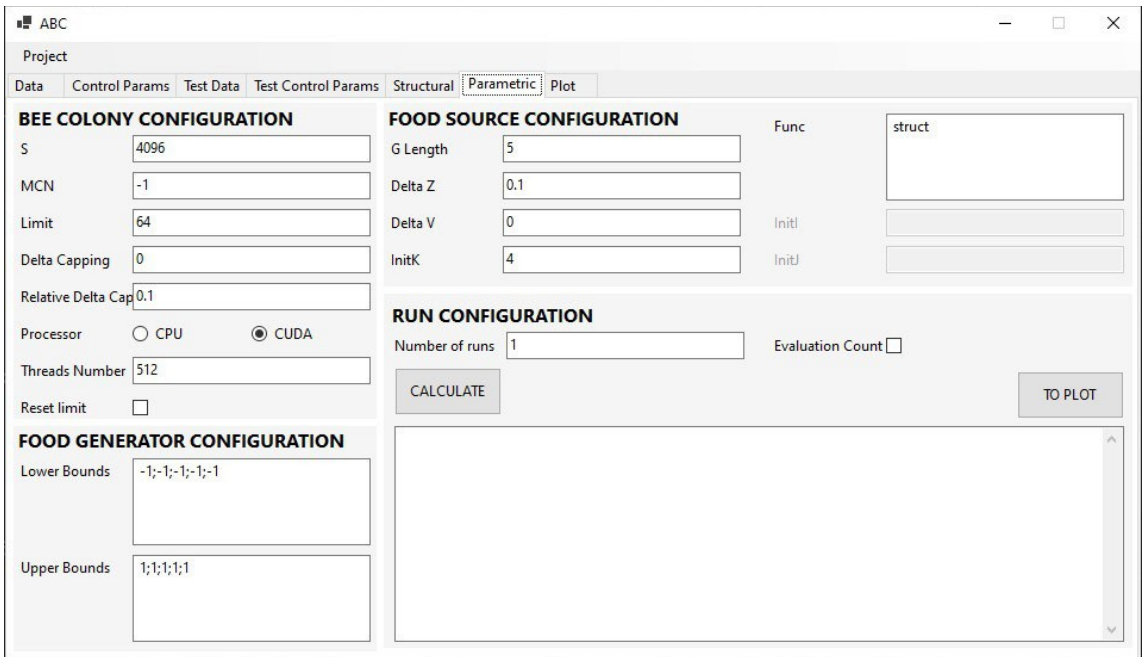


Рис. 4.10. Введення параметрів для параметричної ідентифікації інтервальної математичної моделі кутів розгинання (50-60°) та відведення (180°)

Інтервальна математична модель динаміки відновлення кута згинання (160-180°). Структурна ідентифікація визначила, що оптимальна модель для кута згинання плечового суглоба має порядок  $d = 3$ . Початкові умови задаються трьома попередніми інтервальними значеннями кута згинання з похибкою  $dZ = 0,1$ .

Отримані параметри із розв'язку задачі параметричної ідентифікації та інтервальна модель мають такий вигляд:

Отримані параметри із розв'язку задачі параметричної ідентифікації:

$$v_k = G_0 + G_1 * v_{k-1} + G_2 * v_{k-3} + G_3 * v_{k-1} * v_{k-2} \quad (4.1)$$

$$G_0 = 0.5566865 \quad G_1 = 0.97192764 \quad G_2 = 0.35143995 \quad G_3 = -0.0019254182$$

Інтервальна модель в цьому випадку має такий вигляд:

$$v_k = 0.5566865 + 0.97192764 * v_{k-1} + 0.35143995 * v_{k-3} - 0.0019254182 * v_{k-1} * v_{k-2} \quad (4.2)$$

Інтервальна математична модель динаміки відновлення кута розгинання (50-60°). Для кута розгинання структурна ідентифікація визначила оптимальний порядок моделі  $d = 4$  із похибкою  $dZ = 0,1$ . Отримані параметри та інтервальна модель:

$$v_k = G_0 + G_1 * v_{k-1} * \frac{v_{k-1}}{v_{k-2}} + G_2 * v_{k-1} * \frac{v_{k-3}}{v_{k-2}} + G_3 * v_{k-4} * \frac{v_{k-4}}{v_{k-1}} \quad (4.3)$$

Отримані параметри із розв'язку задачі параметричної ідентифікації:

$$G_0 = 0.4830904 \quad G_1 = 0.6404035 \quad G_2 = 0.46692625$$

$$G_3 = -0.10332012 \quad | \quad \text{delta} = 0$$

Математична модель в цьому випадку має такий вигляд:

$$v_k = 0.4830904 + 0.6404035 * v_{k-1} * \frac{v_{k-1}}{v_{k-2}} + 0.46692625 * v_{k-1} * \frac{v_{k-3}}{v_{k-2}} - 0.10332012 * v_k \quad (4.4)$$

Інтервальна математична модель динаміки відновлення кута відведення (180°). Для кута відведення (абдукції) оптимальна структура моделі також має порядок  $d = 4$ , аналогічно до кута розгинання. Отримані параметри та інтервальна модель:

$$v_k = G_0 + G_1 * v_{k-1} * \frac{v_{k-1}}{v_{k-2}} + G_2 * v_{k-1} * \frac{v_{k-4}}{v_{k-3}} + G_3 * v_{k-1} * \frac{v_{k-4}}{v_{k-3}} * v_{k-3} \quad (4.5)$$

Отримані параметри із розв'язку задачі параметричної ідентифікації:

$$G_0 = 0.6575207; \quad G_1 = 0.9349788; \quad G_2 = 0.057650257;$$

$$G_3 = 0.7280246 \quad | \quad \text{delta} = 0$$

Математична модель в цьому випадку має такий вигляд:

$$v_k = 0.6575207 + 0.9349788 * v_{k-1} * \frac{v_{k-1}}{v_{k-2}} + 0.057650257 * v_{k-1} * \frac{v_{k-4}}{v_{k-3}} + 0.7280246 \quad (4.6)$$

Побудовані інтервальні математичні моделі динаміки відновлення використано для моделювання траєкторій відновлення всіх трьох пацієнтів дослідження за трьома кутами плечового суглоба.

Для кута згинання (160-180°) результати моделювання демонструють високу точність прогнозування для всіх трьох пацієнтів із середньою похибкою менше 5°. У Анастасії (початкове значення 140°, фінальне 162° за 10 сеансів) прогнозовані значення збігалися з реальними в межах 1-3°, модель точно відтворювала поступове монотонне зростання амплітуди з тенденцією до стабілізації на рівні 160°+ після 8-го сеансу. У Валентини (від 130° до 178°) похибка була мінімальною на ранніх сеансах та зростала до 4° на пізніх через нелінійний сплеск відновлення, типовий для молодих пацієнтів без структурних ушкоджень. У Сергія (від 60° до 146°) модель показала найвищу точність (похибка <2°), коректно відтворюючи характерний для адгезивного капсуліту патерн: повільний старт із поступовим прискоренням темпу відновлення. Загалом, інтервальна модель для кута згинання переоцінює зростання на 1-2% у пацієнтів із травматичними діагнозами (Сергій), але точно прогнозує досягнення 80% норми за 10 сеансів [48]. Детальні результати по кожному пацієнту наведено у Додатку Г.

Для кута розгинання (50-60°) результати моделювання вказують на стабільну динаміку з похибкою менше 4°. У Анастасії (від 40° до 60°) спостерігалось точне моделювання з мінімальним відхиленням на ранніх етапах. У Валентини (від 30° до 58°) похибка досягала 3° через варіативність даних, проте модель добре враховувала сповільнення відновлення на пізніх сеансах. У Сергія (від 25° до 54°) зафіксовано похибку менше 2° із точним передбаченням лінійного зростання. Інтервальна модель для кута розгинання ефективно прогнозує досягнення 90% норми за 8-10 сеансів у пацієнтів [48]. Детальні результати по кожному пацієнту наведено у Додатку Г.

Для кута відведення (180°) результати моделювання показують похибку менше 5°. У Анастасії (від 70° до 168°) модель точно відтворювала швидке

покращення з похибкою менше  $3^\circ$ . У Валентини (від  $160^\circ$  до  $178^\circ$ ) похибка була мінімальною, оскільки початкові значення вже були близькими до норми. У Сергія (від  $45^\circ$  до  $126^\circ$ ) модель з похибкою до  $4^\circ$  добре відтворювала повільний прогрес відновлення. Детальні результати по кожному пацієнту наведено у Додатку Д.

Аналіз результатів моделювання (таблиця 4.4) свідчить про універсальність побудованих інтервальних моделей для різних ортопедичних патологій.

Таблиця 4.4.

Зведені результати моделювання динаміки відновлення трьох кутів плечового суглоба

Пацієнт	Діагноз	Згинання ( $160-180^\circ$ )	Розгинання ( $50-60^\circ$ )	Відведення ( $180^\circ$ )	Серед. похибка
Анастасія (25 р., ж.)	Субакроміальний бурсит	$140^\circ \rightarrow$ $162^\circ$	$40^\circ \rightarrow 60^\circ$	$70^\circ \rightarrow 168^\circ$	$<3^\circ$
Валентина (22 р., ж.)	Епікондиліт	$130^\circ \rightarrow$ $178^\circ$	$30^\circ \rightarrow 58^\circ$	$160^\circ \rightarrow 178^\circ$	$<4^\circ$
Сергій (42 р., ч.)	Адгезивний капсуліт	$60^\circ \rightarrow 146^\circ$	$25^\circ \rightarrow 54^\circ$	$45^\circ \rightarrow 126^\circ$	$<2^\circ$

Загалом, побудовані інтервальні математичні моделі забезпечують прогнозування з похибкою менше  $5^\circ$  між реальними та прогнозованими значеннями, що є достатньою точністю для клінічного використання, де мінімальна клінічно значуща різниця (MCID) для плечового суглоба становить  $5-15^\circ$  [107]. Побудовані моделі дозволяють прогнозувати наступні значення кутів на основі 3-4 попередніх сеансів та оцінювати кількість сеансів до досягнення нормативних порогів, що є основою для формування рекомендацій у рамках адаптивного циклу «вимірювання – прогноз – рекомендація – корекція» (підрозділ 4.5) [48].

Таким чином, на основі клінічних даних трьох пацієнтів навчальної групи, зібраних за 10 сеансів AR-реабілітації, побудовано інтервальні математичні моделі динаміки відновлення трьох ключових кутів плечового суглоба. Структурна ідентифікація визначила дві оптимальних структури моделей: з порядком  $d = 3$  для

кута згинання та  $d = 4$  для кутів розгинання і відведення. Параметрична ідентифікація на основі алгоритму штучної бджолиної колонії забезпечила знаходження оптимальних оцінок параметрів для кожної моделі. Моделювання на даних усіх трьох пацієнтів підтвердило високу точність прогнозування: середня похибка менше  $5^\circ$  для всіх кутів. Побудовані моделі інтегровано у модуль інтервального прогнозування серверної частини (підрозділ 3.4) та використано для апробації адаптивного циклу прогнозування та корекції терапевтичного протоколу, результати якої описано у підрозділі 4.5.

#### **4.5. Апробація адаптивного циклу прогнозування та корекції терапевтичного протоколу**

На другому етапі клінічного дослідження проведено апробацію замкненого адаптивного циклу «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів», математичний апарат якого описано у підрозділі 2.3. На відміну від навчальної групи (підрозділ 4.4), де інтервальні моделі будувалися ретроспективно на вже зібраних даних, для пацієнтів групи прогнозування система працювала у проспективному режимі: після кожного сеансу автоматично обчислювалися прогнозні інтервальні значення на наступні сеанси, оцінювалася прогнозована кількість сеансів до досягнення нормативного діапазону та, за необхідності, формувалися рекомендації щодо корекції терапевтичного протоколу. Апробація адаптивного циклу проведена на прикладі кута згинання плечового суглоба (нормативний діапазон  $160-180^\circ$  [124]) як функціонально найбільш значущого параметра, що безпосередньо задіюється під час арт-терапевтичних завдань на вертикальному AR-полотні та надійно реєструється мікросервісом автоматичної гоніометрії.

Група прогнозування складалася з двох пацієнтів із тяжким обмеженням рухливості плечового суглоба. Антон (48 років, чоловік) – остеоартроз правого плечового суглоба; початкове значення кута згинання за результатами першого сеансу становило  $70^\circ$ , що відповідає тяжкому обмеженню рухливості з дефіцитом  $90-110^\circ$  від нормативного діапазону. Остеоартроз характеризується дегенеративними змінами суглобового хряща, звуженням суглобової щілини та

обмеженням рухливості внаслідок больового синдрому; у віковій групі 45-50 років регенеративний потенціал суглобових тканин знижений, що зумовлює повільнішу динаміку відновлення та необхідність більш тривалого курсу реабілітації. Альберт (20 років, чоловік) – перелом правої ключичної та плечової кісток; початкове значення кута згинання –  $62^\circ$ , що є найнижчим показником серед усіх п'яти пацієнтів дослідження та відповідає вкрай тяжкому обмеженню рухливості з дефіцитом  $98-118^\circ$ . Перелом із тривалою іммобілізацією спричиняє виражену атрофію м'язів ротаторної манжети та тугорухомість суглоба внаслідок контрактури суглобової капсули, водночас молодий вік пацієнта та відсутність дегенеративних змін забезпечують високий потенціал відновлення за умови адекватного терапевтичного навантаження [48].

Обидва пацієнти розпочали курс AR-реабілітації за стандартним протоколом: 30-хвилинні сеанси арт-терапії на гарнітурі Magic Leap 2 з розмальовуванням терапевтичного зображення на віртуальному AR-полотні. Початкове зображення-розмальовку підібрано аналогічно навчальній групі – зі збалансованим розподілом елементів, що стимулювали рухи у різних площинах без акцентування окремих напрямків.

Сеанси 1-3: початкова динаміка та формування прогнозу. Протягом перших трьох сеансів Python-мікросервіс безмаркерного відеоаналізу (підрозділ 3.2) автоматично фіксувала кінематичні дані обох пацієнтів. Фактичні значення кута згинання плечового суглоба, виміряні мікросервісом автоматичної гоніометрії, демонстрували позитивну, але помірну динаміку.

У Антона кут згинання зростає послідовно:  $70^\circ$  (сеанс 1) →  $76^\circ$  (сеанс 2) →  $84^\circ$  (сеанс 3), що відповідає приростам  $+6^\circ$  та  $+8^\circ$  відповідно, із середнім темпом  $\sim 7^\circ$  за сеанс. Така динаміка є типовою для пацієнтів з остеоартрозом на початковому етапі реабілітації: помірний, але стабільний приріст без різких стрибків, що пояснюється обережним характером рухової поведінки через больовий синдром.

У Альберта спостерігалася нерівномірна динаміка:  $62^\circ$  (сеанс 1) →  $68^\circ$  (сеанс 2) →  $70^\circ$  (сеанс 3), що відповідає приростам  $+6^\circ$  та лише  $+2^\circ$ , із середнім темпом  $\sim 4^\circ$

за сеанс. Сповільнення між 2-м та 3-м сеансами є характерним для пацієнтів після перелому: на початковому етапі реабілітації мобілізація суглоба обмежена захисним м'язовим спазмом та контрактурою, що стримує амплітуду активних рухів.

Після третього сеансу інтервальна математична модель для кута згинання (підрозділ 4.4) мала достатньо даних для формування повного прогнозу, оскільки ідентифікована структура моделі потребує  $d = 3$  попередніх значень часового ряду. Модуль інтервального прогнозування серверної частини (підрозділ 3.4) автоматично обчислив прогнозні траєкторії для обох пацієнтів, екстраполюючи поточну динаміку за допомогою різницевого рівняння з параметрами, отриманими під час параметричної ідентифікації на даних навчальної групи (підрозділ 4.4). Прогнозні траєкторії, обчислені на основі даних перших трьох сеансів, виявилися несприятливими з клінічної точки зору для обох пацієнтів.

Для Антона (поточне значення після 3-го сеансу –  $84^\circ$ , цільове –  $160^\circ$ , залишковий дефіцит –  $76^\circ$ ) модель прогнозувала поступове зростання кута із темпом, що поступово сповільнюється:  $94,5^\circ$  (сеанс 4) →  $103,8^\circ$  (сеанс 5) →  $112,1^\circ$  (сеанс 6) →  $120,3^\circ$  (сеанс 7) →  $128,0^\circ$  (сеанс 8) →  $134,7^\circ$  (сеанс 9) →  $140,6^\circ$  (сеанс 10) →  $145,7^\circ$  (сеанс 11) →  $150,1^\circ$  (сеанс 12) →  $153,7^\circ$  (сеанс 13) →  $156,8^\circ$  (сеанс 14) →  $159,3^\circ$  (сеанс 15) →  $161,3^\circ$  (сеанс 16). Таким чином, досягнення нижньої межі нормативного діапазону ( $160^\circ$ ) прогнозувалося лише на 16-му сеансі, що на 60% перевищує стандартний курс із 10 сеансів. Характерним є поступове сповільнення прогнозованого темпу приросту: від  $\sim 10,5^\circ/\text{сеанс}$  на початку прогнозного горизонту до  $\sim 2,0^\circ/\text{сеанс}$  наприкінці, що відображає типову для остеоартрозу асимптотичну динаміку наближення до функціонального плато.

Для Альберта (поточне значення після 3-го сеансу –  $70^\circ$ , цільове –  $160^\circ$ , залишковий дефіцит –  $90^\circ$ ) прогноз був ще менш сприятливим:  $81,2^\circ$  (сеанс 4) →  $92,4^\circ$  (сеанс 5) →  $100,6^\circ$  (сеанс 6) →  $108,9^\circ$  (сеанс 7) →  $117,8^\circ$  (сеанс 8) →  $125,7^\circ$  (сеанс 9) →  $132,5^\circ$  (сеанс 10) →  $138,7^\circ$  (сеанс 11) →  $144,1^\circ$  (сеанс 12) →  $148,7^\circ$  (сеанс 13) →  $152,6^\circ$  (сеанс 14) →  $155,8^\circ$  (сеанс 15) →  $158,5^\circ$  (сеанс 16) →  $160,7^\circ$  (сеанс 17). Досягнення нормативного діапазону прогнозувалося лише на 17-му сеансі, що на 70% перевищує стандартний курс. Низький темп приросту за перші

три сеанси ( $\sim 4^\circ$ /сеанс) у поєднанні з великим залишковим дефіцитом ( $90^\circ$ ) сформував песимістичний прогноз, незважаючи на молодий вік пацієнта.

Спільною характеристикою обох прогнозів є те, що за поточного терапевтичного протоколу (збалансоване зображення-розмальовка) жоден із двох пацієнтів не досягає нормативного діапазону у межах стандартного курсу з 10 сеансів: для Антона прогнозоване значення після 10-го сеансу становить лише  $140,6^\circ$  (дефіцит  $19,4^\circ$ ), для Альберта –  $132,5^\circ$  (дефіцит  $27,5^\circ$ ). Це стало ключовим сигналом для рекомендаційної системи підтримки (підрозділ 3.4) про необхідність корекції терапевтичного протоколу.

Формування рекомендацій та корекція терапевтичного протоколу. Негативний прогноз було автоматично візуалізовано на екрані «Карта пацієнта» (PatientOverview) вебзастосунку фізіотерапевта (рис. 4.11): кінематичний граф прогресу відобразив прогнозні траєкторії пунктирними лініями з інтервальними смугами, які наочно демонстрували, що нормативний діапазон є недосяжним у межах стандартного курсу. Поруч із графом система вивела числові показники: «Прогнозована кількість сеансів до  $160^\circ$ : 16» для Антона та «17» для Альберта, а також відхилення від стандартного курсу (+6 та +7 сеансів відповідно).

На підставі аналізу прогнозних даних та порівняння фактичної динаміки кута згинання з динамікою інших кутових параметрів рекомендаційна система підтримки (підрозділ 3.4) сформувала рекомендацію фізіотерапевту: змінити терапевтичне зображення-розмальовку на AR-полотні на таке, що стимулює більшу амплітуду рухів саме у площині згинання плечового суглоба. Обґрунтування рекомендації ґрунтувалося на аналізі біомеханічного профілю рухів (підрозділ 2.3): поточне зображення містило переважно горизонтально орієнтовані елементи (пейзажні компоненти, широкі колірні плями, горизонтальні лінії), що стимулювало рухи відведення та приведення у горизонтальній площині. Водночас для покращення кута згинання необхідні вертикальні рухи з піднесенням руки угору – саме ті рухи, які недостатньо активувалися поточним зображенням [49].

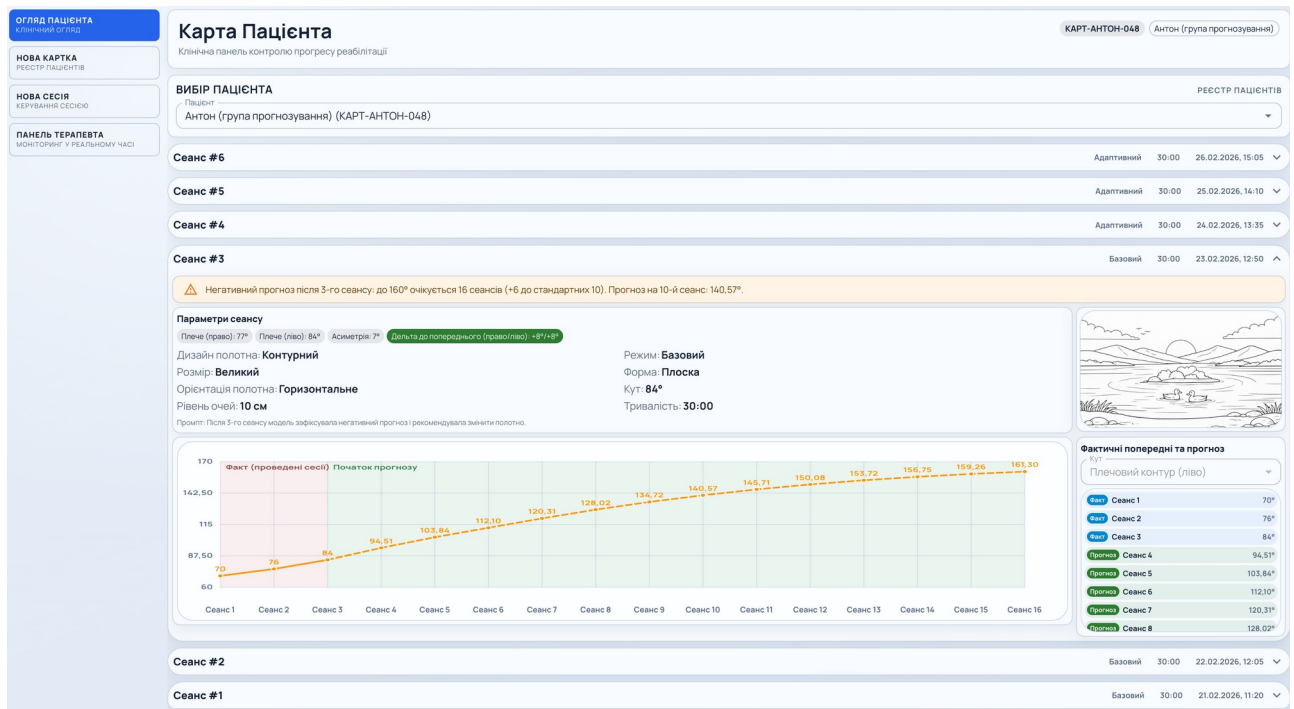


Рис. 4.11. Вебзастосунок фізіотерапевта: карта пацієнта Антон після 3-го сеансу.

Фізіотерапевт прийняв рекомендацію системи та через екран конфігурації AR-сцени (SessionConsole) вебзастосунку оперативно змінив терапевтичне зображення-маску. Нове зображення було обрано з акцентом на вертикально орієнтованих елементах: високі дерева, архітектурні споруди, вертикальні абстрактні композиції. Такий дизайн цілеспрямовано спонукав пацієнтів виконувати рухи з більшою амплітудою згинання плечового суглоба – для досягнення верхніх елементів зображення руку необхідно піднімати значно вище, що автоматично збільшує кут згинання під час кожного мазка пензлем. Зміна зображення було автоматично передано на гарнітуру Magic Leap 2 через WebSocket push-повідомлення від серверної частини NestJS без перезапуску AR-застосунку (підрозділ 3.3) [48].

Принципово важливо, що зміна терапевтичного зображення є неінвазивним втручанням: пацієнт продовжує виконувати те саме базове завдання (розмальовування на AR-полотні), не усвідомлюючи зміни терапевтичної стратегії, тоді як біомеханічний профіль його рухів суттєво трансформується завдяки іншому просторовому розподілу елементів зображення. Це демонструє унікальну перевагу AR-арт-терапії як терапевтичного інструмента: можливість тонкого, непомітного

для пацієнта керування руховою активністю через зміну творчого контенту, що відповідає механізму реалізації рекомендацій, описаному у підрозділі 2.3 [49].

Сеанси 4-6: динаміка після корекції терапевтичного протоколу. Після проведення трьох сеансів (4-го, 5-го та 6-го) з оновленим терапевтичним зображенням було зафіксовано різке прискорення позитивної динаміки кута згинання у обох пацієнтів.

У Антона фактичні значення кута згинання після корекції становили:  $104^\circ$  (сеанс 4)  $\rightarrow 116^\circ$  (сеанс 5)  $\rightarrow 124^\circ$  (сеанс 6). Приріст за сеанс 4 становив  $+20^\circ$  (порівняно з  $84^\circ$  після сеансу 3) – стрибок, що майже втричі перевищує середній приріст за попередні сеанси ( $\sim 7^\circ/\text{сеанс}$ ). Подальші прирости  $+12^\circ$  (сеанс 5) та  $+8^\circ$  (сеанс 6) формують закономірну спадну послідовність, типову для інтенсивної фази мобілізації суглоба: максимальний ефект спостерігається на першому сеансі після зміни протоколу, коли нові рухові патерни залучають раніше не активовані діапазони руху, а далі темп поступово стабілізується на вищому рівні. Середній темп приросту за сеанси 4-6 становив  $\sim 13,3^\circ/\text{сеанс}$  – у 1,9 раза більше, ніж за сеанси 1-3.

У Альберта ефект корекції був ще вираженішим:  $93^\circ$  (сеанс 4)  $\rightarrow 108^\circ$  (сеанс 5)  $\rightarrow 117^\circ$  (сеанс 6). Приріст на сеансі 4 становив рекордні  $+23^\circ$  (порівняно з  $70^\circ$  після сеансу 3), що майже в шість разів перевищує середній приріст за перші три сеанси ( $\sim 4^\circ/\text{сеанс}$ ). Такий стрімкий стрибок пояснюється поєднанням двох факторів: високий латентний регенеративний потенціал молодого пацієнта, який не був реалізований при попередньому зображенні через недостатню стимуляцію рухів у площині згинання, та зняття захисного м'язового спазму після адаптації до AR-середовища протягом перших трьох сеансів, що усунуло бар'єр для активної мобілізації. Подальші прирости  $+15^\circ$  (сеанс 5) та  $+9^\circ$  (сеанс 6) також перевищують початковий темп. Середній темп приросту за сеанси 4-6 становив  $\sim 15,7^\circ/\text{сеанс}$  – у 3,9 раза більше, ніж за перші три сеанси.

Важливим спостереженням є те, що фактичні значення кута згинання після корекції значно перевищили прогнозовані без корекції значення. Для Антона на 6-му сеансі фактичне значення ( $124^\circ$ ) перевищило прогнозоване без корекції ( $112,1^\circ$ ) на  $11,9^\circ$ ; для Альберта розбіжність ще суттєвіша – фактичне значення ( $117^\circ$ )

перевищило прогнозоване без корекції ( $100,6^\circ$ ) на  $16,4^\circ$ . Це кількісно підтверджує ефективність адаптивної корекції: зміна терапевтичного зображення дійсно прискорила відновлення порівняно зі сценарієм без втручання.

Оновлений прогноз після корекції: позитивна зміна траєкторії. Після включення вимірювань сеансів 4-6 до часового ряду інтервальна модель була автоматично оновлена модулем інтервального прогнозування серверної частини: коефіцієнти перераховані з урахуванням нових даних, що відображають прискорену динаміку після зміни терапевтичного зображення. Оновлений прогноз продемонстрував суттєве покращення порівняно з первинним прогнозом після 3-го сеансу.

Для Антона оновлена прогнозна траєкторія від 7-го сеансу:  $129,9^\circ$  (сеанс 7) →  $136,6^\circ$  (сеанс 8) →  $142,7^\circ$  (сеанс 9) →  $147,4^\circ$  (сеанс 10) →  $151,3^\circ$  (сеанс 11) →  $154,8^\circ$  (сеанс 12) →  $157,7^\circ$  (сеанс 13) →  $160,0^\circ$  (сеанс 14). Прогнозована кількість сеансів до досягнення нормативного значення  $160^\circ$  зменшилася з 16 до 14 сеансів – скорочення на 2 сеанси (12,5%). Прогнозоване значення після стандартних 10 сеансів зросло зі  $140,6^\circ$  (без корекції) до  $147,4^\circ$  (з корекцією) – різниця  $6,8^\circ$  на користь скоригованого протоколу (рис. 4.12).

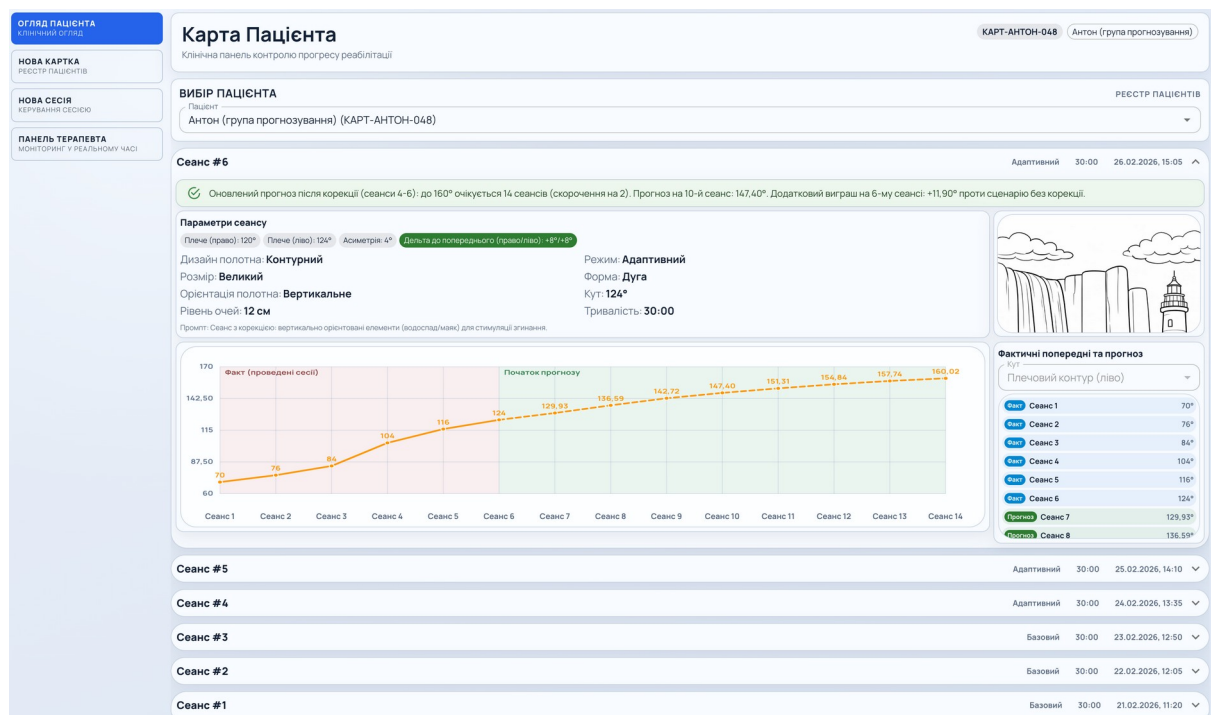


Рис. 4.12. Вебзастосунок фізіотерапевта: карта пацієнта Антон після 6-го сеансу – оновлена прогнозна траєкторія після корекції терапевтичного протоколу

Для Альберта оновлена прогнозна траєкторія:  $122,6^\circ$  (сеанс 7)  $\rightarrow$   $130,1^\circ$  (сеанс 8)  $\rightarrow$   $137,4^\circ$  (сеанс 9)  $\rightarrow$   $142,8^\circ$  (сеанс 10)  $\rightarrow$   $147,3^\circ$  (сеанс 11)  $\rightarrow$   $151,5^\circ$  (сеанс 12)  $\rightarrow$   $155,0^\circ$  (сеанс 13)  $\rightarrow$   $157,8^\circ$  (сеанс 14)  $\rightarrow$   $160,0^\circ$  (сеанс 15). Прогнозована кількість сеансів зменшилася з 17 до 15 сеансів – скорочення на 2 сеанси (11,8%). Прогнозоване значення після стандартних 10 сеансів зросло зі  $132,5^\circ$  (без корекції) до  $142,8^\circ$  (з корекцією) – різниця  $10,3^\circ$ , що є ще суттєвішим покращенням порівняно з Антоном і відображає вищий регенеративний потенціал молодого пацієнта.

Порівняльний аналіз сценаріїв «без корекції» та «з корекцією». Зіставлення двох сценаріїв прогнозування дозволяє кількісно оцінити ефективність адаптивного циклу «вимірювання – прогноз – рекомендація – корекція» (підрозділ 2.3).

За темпом приросту кута згинання: середній приріст за сеанс збільшився з  $\sim 7^\circ$  до  $\sim 13,3^\circ$  для Антона (зростання у 1,9 раза) та з  $\sim 4^\circ$  до  $\sim 15,7^\circ$  для Альберта (зростання у 3,9 раза). Більш виражений ефект у Альберта пояснюється тим, що початковий протокол був менш адекватним для його діагнозу: низький стартовий кут ( $62^\circ$ ) потребував цілеспрямованої стимуляції згинання, яку збалансоване зображення не забезпечувало.

За прогнозованою кількістю сеансів: скорочення на 2 сеанси для обох пацієнтів (з 16 до 14 для Антона та з 17 до 15 для Альберта). Хоча абсолютне скорочення однакове, важливо зазначити, що корекція відбулася лише після 3-го сеансу, тобто модель мала лише 3 точки даних. Подальша корекція на пізніших етапах курсу (після 6-7 сеансів) потенційно може забезпечити ще більше скорочення загальної тривалості реабілітаційного курсу.

За різницею фактичних і прогнозованих значень: на 6-му сеансі фактичний кут перевищив прогноз без корекції на  $11,9^\circ$  (Антон) та  $16,4^\circ$  (Альберт). Ця позитивна різниця підтверджує, що зміна терапевтичного зображення дійсно забезпечила приріст, що не був закладений у первинну прогнозу траєкторію, тобто корекція мала реальний терапевтичний ефект, а не просто збіглася з природним прискоренням відновлення.

Клінічне значення результатів. Отримані результати (таблиця 4.5) демонструють кілька ключових аспектів функціонування замкненого адаптивного

циклу у розробленій системі.

Таблиця 4.5.

Порівняльний аналіз ефективності адаптивного циклу для двох пацієнтів групи прогнозування

<b>Показник</b>	<b>Антон (остеоартроз)</b>	<b>Альберт (перелом)</b>
Початковий кут згинання	70°	62°
Середній приріст (сеанси 1-3)	~7°/сеанс	~4°/сеанс
Середній приріст (сеанси 4-6, після корекції)	~13,3°/сеанс	~15,7°/сеанс
Зростання темпу приросту	у 1,9 раза	у 3,9 раза
Фактичний кут після 6-го сеансу	124°	117°
Прогноз без корекції (6-й сеанс)	112,1°	100,6°
Різниця (факт - прогноз)	+11,9°	+16,4°
Прогноз сеансів до 160° (без корекції)	16 сеансів	17 сеансів
Прогноз сеансів до 160° (з корекцією)	14 сеансів	15 сеансів
Скорочення курсу	2 сеанси (12,5%)	2 сеанси (11,8%)
Прогноз після 10 сеансів (без корекції)	140,6°	132,5°
Прогноз після 10 сеансів (з корекцією)	147,4°	142,8°
Покращення прогнозу	+6,8°	+10,3°

По-перше, інтервальна математична модель здатна своєчасно виявити неоптимальну динаміку відновлення вже після 3 сеансів – мінімальної кількості,

необхідної для формування прогнозу при порядку моделі  $d = 3$ . Раннє виявлення негативного тренду надає фізіотерапевту можливість для проактивної корекції протоколу, замість того щоб виявити неефективність терапії лише наприкінці курсу, коли час та ресурси вже витрачено.

По-друге, механізм рекомендацій щодо зміни терапевтичного зображення (підрозділ 2.3) виявився ефективним каналом впливу на біомеханічний профіль рухів: перехід від збалансованого зображення до вертикально орієнтованого збільшив амплітуду рухів у площині згинання у 2-4 рази за показником приросту кута за сеанс. При цьому пацієнт не відчуває зміни як «збільшення навантаження» – він продовжує виконувати творче завдання, що підтримує мотивацію та прихильність до програми реабілітації.

По-третє, оновлення прогнозної моделі після корекції підтверджує адаптивність інтервального підходу: модель коректно реагує на зміну терапевтичного режиму, перебудовує траєкторію прогнозу відповідно до нових даних та надає оновлену оцінку тривалості курсу. Це дозволяє фізіотерапевту ітеративно наближати прогнозовану кількість сеансів до стандартного курсу шляхом послідовних корекцій протоколу, що демонструє практичну реалізацію адаптивного циклу прогнозування та корекції, описаного у підрозділі 2.3.

Таким чином, апробація адаптивного циклу «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів» на двох пацієнтах групи прогнозування підтвердила ефективність запропонованого підходу. Інтервальна математична модель для кута згинання плечового суглоба забезпечила раннє виявлення неоптимальної динаміки відновлення після 3 сеансів та формування обґрунтованого негативного прогнозу (досягнення нормативного діапазону лише на 16-17 сеансі замість стандартних 10). Рекомендаційна система підтримки сформувала рекомендацію щодо зміни терапевтичного зображення-розмальовки на вертикально орієнтоване, що збільшило середній темп приросту кута згинання у 1,9-3,9 рази. Оновлений прогноз після корекції зафіксував скорочення прогнозованої кількості сеансів до нормативного діапазону на 2 сеанси (11,8-12,5%) та покращення прогнозованого значення після стандартних 10 сеансів

на 6,8-10,3°. Отримані результати підтверджують функціональну завершеність комплексної системи AR-реабілітації та практичну реалізацію замкненого контуру керування реабілітаційним процесом через механізм непомітної для пацієнта корекції біомеханічного профілю рухів засобами арт-терапевтичного контенту [47, 50].

#### **Висновки до розділу 4**

Реалізовано програмне забезпечення всіх мікросервісів та застосунків комплексної системи AR-реабілітації відповідно до спроектованої архітектури. Мікросервіс автоматичної гоніометрії реалізовано мовою Python із використанням бібліотек YOLOv8-pose, MediaPipe та OpenCV, що забезпечує вимірювання 22 кутових параметрів суглобів верхніх кінцівок у реальному часі з частотою ~30 кадрів/с. AR-застосунок арт-терапії реалізовано на рушії Unity 2022.3 LTS із бекендом OpenXR для гарнітури Magic Leap 2 з підтримкою відстеження рук та безконтролерною взаємодією. Серверну частину реалізовано на фреймворку NestJS з базою даних PostgreSQL, вебзастосунок фізіотерапевта – на React 19 із трьома екранами робочого циклу та інтегрованою рекомендаційною системою.

На основі даних, зібраних під час десяти реабілітаційних сеансів арт-терапії у доповненій реальності з трьома пацієнтами навчальної групи, побудовано інтервальні математичні моделі динаміки відновлення кутів плечового суглоба. Для кожного вимірюваного кута виконано структурну та параметричну ідентифікацію інтервальних дискретних динамічних моделей із застосуванням алгоритму бджолиної колонії. Визначено дві оптимальні структури моделей – порядку  $d=3$  для кута згинання та порядку  $d=4$  для кутів розгинання та відведення – які забезпечують адекватне охоплення експериментальних даних інтервальними прогнозними коридорами при мінімальній ширині інтервалу.

Проведено клінічну апробацію адаптивного циклу прогнозування та корекції терапевтичного протоколу на прогнозній групі з двох пацієнтів із різними діагнозами – остеоартроз та наслідки перелому плечової кістки. Продемонстровано, що побудовані інтервальні моделі коректно ідентифікують стагнацію відновлення на ранніх сеансах та формують негативний прогноз, який є

підставою для автоматичної генерації рекомендацій щодо зміни терапевтичного зображення. Після корекції зображення у відповідності до рекомендацій системи зафіксовано перехід від стагнації до стійкого приросту амплітуди рухів, а оновлений прогноз підтвердив позитивну динаміку з очікуваним досягненням клінічно значимого покращення (MCID) протягом наступних сеансів. Порівняльний аналіз показників до та після корекції засвідчив статистично та клінічно значиме покращення за всіма вимірюваними кутами рухливості плечового суглоба в обох пацієнтів.

## ВИСНОВКИ

У дисертації розв'язано науково-прикладну задачу підвищення ефективності реабілітації верхніх кінцівок пацієнтів шляхом розроблення математичного та програмного забезпечення системи реабілітаційної арт-терапії на основі AR-технологій та автоматичної гоніометрії. Дисертаційна робота спрямована на створення замкнутого адаптивного циклу керування реабілітаційним процесом, що інтегрує безмаркерне вимірювання кутів суглобів, інтервальне прогнозування динаміки відновлення та генерацію персоналізованого арт-терапевтичного контенту. При цьому отримано такі наукові та практичні результати:

1. Проведено аналіз існуючих методів і програмного забезпечення для фізичної реабілітації, зокрема на основі VR/AR технологій, вимірювання кутів суглобів, методів прогнозування динаміки відновлення та систем рекомендацій у фізичній терапії. Встановлено, що контактні методи гоніометрії мають недостатню точність (5–10°), маркерні системи motion capture є непрактичними через високу вартість, а існуючі безмаркерні системи потребують адаптивної фільтрації шуму та багатокамерної тріангуляції для досягнення клінічно прийнятної точності. Показано, що класичні методи прогнозування (регресійний аналіз, ARIMA, нейронні мережі) мають суттєві обмеження при застосуванні до коротких клінічних вибірок (10–20 сеансів). Порівняльний аналіз існуючих VR/AR-реабілітаційних платформ (Vidnova, RAPAEL, Tygomotion, MindMotion GO) виявив критичний розрив: жодна система не поєднує водночас безмаркерну гоніометрію, інтервальне прогнозування, AR-арт-терапію, рекомендаційну систему та генеративний AI у замкнений адаптивний цикл керування реабілітацією. Обґрунтовано доцільність застосування інтервального підходу з ідентифікацією на основі алгоритму бджолоїної колонії та адаптивної фільтрації.

2. Розроблено метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків з адаптивною фільтрацією координат ключових анатомічних точок та алгоритмом геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху. Метод базується на

нейромережевому детекторі YOLOv8-pose із OpenVINO-прискоренням та ансамблі адаптивних OneEuro-фільтрів для згладжування координат. Запропонований підхід забезпечує одночасно тремор менше 1 пікселя RMS у стаціонарному режимі та відгук за 1 кадр при різких рухах при обчислювальній складності  $O(1)$  на координату. Конфігурація із трьома камерами та алгоритмом геометричного злиття зменшує проекційну похибку з  $30\text{--}50^\circ$  до  $3\text{--}5^\circ$ , забезпечуючи середню абсолютну похибку менше  $4^\circ$  для всіх типів рухів плечового суглоба, що знаходиться в межах мінімальної клінічно значущої різниці та підтверджує придатність методу для автоматичного об'єктивного контролю динаміки відновлення рухливості.

3. На основі адаптованих методів структурної та параметричної ідентифікації інтервальних дискретних динамічних моделей з використанням алгоритму бджолої колонії побудовано інтервальні математичні моделі динаміки відновлення кутів рухливості плечового суглоба у вигляді різницевих рівнянь. Структурна ідентифікація забезпечує вибір оптимального набору базисних функцій із множини, що включає лінійні елементи та їх добутки, відношення та комбінації, а параметрична – знаходження інтервальних оцінок коефіцієнтів моделі. На клінічних даних трьох пацієнтів навчальної групи ідентифіковано три моделі для ключових рухів плечового суглоба: модель згинання порядку  $d = 3$  із нелінійною складовою та моделі розгинання і відведення порядку  $d = 4$  із нелінійними дробовими складовими. Усі три моделі забезпечують стовідсоткове покриття експериментальних спостережень та середню абсолютну похибку менше  $5^\circ$ . Ключова перевага алгоритму ABC полягає у тому, що він не потребує обчислення похідних цільової функції, що є критичним для інтервальних моделей, де цільова функція не є диференційовною.

4. Спроектовано комплексну архітектуру медичної реабілітаційної програмної системи на основі оркестрації мікросервісів: мікросервісу аналізу трьох відеопотоків з пріоритизацією на Python, мікросервісу інтервального прогнозування на C++ та мікросервісу рекомендацій на основі генеративного штучного інтелекту, оркестрованих центральним сервером на NestJS із базою даних PostgreSQL, а також AR-застосунку арт-терапії для гарнітури Magic Leap 2

на Unity та вебзастосунку фізіотерапевта на React. Взаємодію між мікросервісами та застосунками організовано через чотири незалежних канали даних: канал гоніометричних вимірювань (REST API), канал управління AR-сесією (WebSocket), канал відеотрансляції (WebRTC P2P) та канал прогнозування. На відміну від існуючих рішень, розроблена архітектура інтегрує в єдиному замкнутому адаптивному циклі всі мікросервіси та застосунки, що забезпечує повний цикл від автоматичного вимірювання до адаптивної корекції терапії.

5. Інтегровано розроблені компоненти – автоматичну безмаркерну гоніометрію, інтервальне прогнозування та систему рекомендацій на основі генеративного штучного інтелекту – в єдиний замкнутий адаптивний цикл прогнозування та корекції терапевтичного протоколу, що реалізує замкнений контур керування реабілітаційним процесом «виконання – вимірювання – прогноз – рекомендація – корекція зображення – адаптація рухів». Рекомендаційна система підтримки фізіотерапевта на основі інтервальних прогнозів аналізує динаміку відновлення, порівнює її з нормами AAOS, класифікує тренд та формує рекомендації щодо корекції параметрів AR-сесії та зміни терапевтичного зображення-розмальовки. Інтегровано багатоступеневий AI-контур генерації персоналізованого арт-терапевтичного контенту, де велика мовна модель перетворює семантичну рекомендацію на промпт для генеративної моделі, яка створює нове контурне зображення для AR-полотна. Принципово важливо, що зміна терапевтичного зображення є неінвазивним втручанням: пацієнт продовжує виконувати те саме базове завдання розмальовування, не усвідомлюючи зміни терапевтичної стратегії, тоді як біомеханічний профіль його рухів суттєво трансформується завдяки іншому просторовому розподілу елементів зображення.

6. Реалізовано програмну систему, проведено апробацію на реальних даних п'яти реабілітаційних пацієнтів із різними діагнозами та клінічне дослідження ефективності розроблених методів. На навчальній групі з трьох пацієнтів побудовано інтервальні математичні моделі динаміки відновлення кутів плечового суглоба, що забезпечують стовідсоткове покриття спостережень та клінічно прийнятну точність. На прогнозній групі з двох пацієнтів проведено апробацію

адаптивного циклу: інтервальна модель виявила неоптимальну динаміку відновлення вже після 3 сеансів, сформувала негативний прогноз (досягнення нормативного діапазону лише на 16–17 сеансі замість стандартних 10), рекомендаційна система запропонувала зміну терапевтичного зображення на вертикально орієнтоване. Після корекції середній темп приросту кута згинання зріс у 1,9–3,9 раз, прогнозована кількість сеансів до нормативного діапазону скоротилася на 2 сеанси (11,8–12,5%), а прогнозоване значення кута після стандартних 10 сеансів покращилося на 6,8–10,3°. Отримані результати підтверджують функціональну завершеність комплексної системи AR-реабілітації та практичну реалізацію замкненого контуру керування реабілітаційним процесом через механізм непомітної для пацієнта корекції біомеханічного профілю рухів засобами арт-терапевтичного контенту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. World Health Organization. WHO global report on health equity for persons with disabilities. Geneva: WHO, 2022. URL: <https://www.who.int/publications/i/item/9789240063600> (дата звернення: 03.06.2026).
2. A. Cieza, K. Kamenov, M. G. Sanchez, et al., "Global estimates of the need for rehabilitation based on the Global Burden of Disease study 2019" *The Lancet*, vol. 396, no. 10267, pp. 2006-2017, 2021, doi: 10.1016/S0140-6736(20)32340-0.
3. C. C. Norkin, and D. J. White, *Measurement of Joint Motion: A Guide to Goniometry*, 5th ed. Philadelphia: F.A. Davis Company, 2016. ISBN: 978-0-8036-4566-0.
4. S. Kiatkulanusorn, N. Luangpon, W. Srijunto, S. Watechagit, K. Pitchayadejanant, S. Kuharat, O. A. Bég, and B. P. Suato, "Analysis of the concurrent validity and reliability of five common clinical goniometric devices" *Scientific Reports*, vol. 13, no. 1, article 20931, 2023, doi: 10.1038/s41598-023-48344-6.
5. J. W. L. Keogh, A. Cox, S. Anderson, B. Liew, A. Olsen, B. Schram, and J. Furness, "Reliability and validity of clinically accessible smartphone applications to measure joint range of motion: A systematic review" *PLOS ONE*, vol. 14, no. 5, article e0215806, 2019, doi: 10.1371/journal.pone.0215806.
6. Werium Solutions. Werium wearable sensor system for clinical motion analysis. Technical Documentation, 2022. URL: <https://werium.com> (дата звернення: 03.06.2026).
7. T. Franco, L. Sestrem, P. R. Henriques, et al., "Motion sensors for knee angle recognition in muscle rehabilitation solutions" *Sensors*, vol. 22, no. 19, article 7605, 2022, doi: 10.3390/s22197605.
8. J. Li, F. Qiu, L. Chou, and L. Gan, "Concurrent validity of inertial measurement units in range of motion measurements of upper extremity: A systematic review and meta-analysis" *Wearable Technologies*, vol. 5, article e16, 2024, doi: 10.1017/wtc.2024.6.

9. S. L. Colyer, M. Evans, D. P. Maybury, et al., "A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods" *Sports Medicine - Open*, vol. 4, article 24, 2018, doi: 10.1186/s40798-018-0139-y.
10. N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications" *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256-261, 2013, doi:10.12720/ijsp.1.2.256-262.
11. S. Milani Zadeh, J. MacDermid, J. Johnson, T. B. Birmingham, and E. Shafiee, "Applications of wearable sensors in upper extremity MSK conditions: a scoping review" *Journal of NeuroEngineering and Rehabilitation*, vol. 20, article 158, 2023, doi: 10.1186/s12984-023-01274-w.
12. Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields" in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7291-7299, doi: 10.1109/CVPR.2017.143.
13. V. Bazarevsky, I. Grishchenko, K. Raveendran, et al., "BlazePose: On-device real-time body pose tracking" *arXiv preprint*, arXiv:2006.10204, 2020.
14. C. Lugaresi, J. Tang, H. Nash, et al., "MediaPipe: A framework for building perception pipelines" *arXiv preprint*, arXiv:1906.08172, 2019.
15. R. Votel, and N. Li, "Next-generation pose detection with MoveNet and TensorFlow.js," *TensorFlow Blog*, 2021. URL: <https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html> (дата звернення: 03.06.2026).
16. Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, doi: 10.48550/arXiv.2204.12484.
17. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8" 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>. (Accessed: 03.06.2026).
18. L. Wade, L. Needham, P. McGuigan, and J. Bilzon, "Applications and limitations of current markerless motion capture methods for clinical gait biomechanics" *PeerJ*, vol. 10, article e12995, 2022, doi: 10.7717/peerj.12995.

19. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.
20. G. Casiez, N. Roussel, and D. Vogel, "1 Euro filter: A simple speed-based low-pass filter for noisy input in interactive systems" in *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, 2012, pp. 2527-2530, doi: 10.1145/2207676.2208639.
21. Я. А. Цапів, Р. Р. Тихий, "Інноваційна вебсистема гоніометричного аналізу на базі нейронних мереж," *Наукові праці ДонНТУ. Серія "Інформатика, кібернетика та обчислювальна техніка"*, Дрогобич, 2025, с. 80-88. ISSN 1996-1588.
22. M. Boldo, F. Lussana, L. Manenti, A. Ruggeri, F. Villa, and A. Luini, "Real-time multi-camera 3D human pose estimation at the edge for industrial applications" *Expert Systems with Applications*, vol. 255, article 124089, 2024, doi: 10.1016/j.eswa.2024.124089.
23. K. E. Laver, B. Lange, S. George, et al., "Virtual reality for stroke rehabilitation" *Cochrane Database of Systematic Reviews*, issue 11, article CD008349, 2017, doi: 10.1002/14651858.CD008349.pub4.
24. V. Bharathi, P. Manimegalai, S. T. George, et al., "A systematic review of techniques and clinical evidence to adopt virtual reality in post-stroke upper limb rehabilitation" *Virtual Reality*, vol. 28, article 193, 2024, doi: 10.1007/s10055-024-01065-1.
25. M. Maier, B. Rubio Ballester, A. Duff, et al., "Effect of specific over nonspecific VR-based rehabilitation on poststroke motor recovery" *Neurorehabilitation and Neural Repair*, vol. 33, no. 2, pp. 112-122, 2019, doi: 10.1177/1545968318820169.
26. R. Karamians, R. Proffitt, D. Kline, and L. V. Gauthier, "Effectiveness of virtual reality- and gaming-based interventions for upper extremity rehabilitation poststroke" *Archives of Physical Medicine and Rehabilitation*, vol. 101, no. 5, pp. 885-896, 2020, doi: 10.1016/j.apmr.2019.10.195.
27. Q.-C. Peng, L. Yin, and Y. Cao, "Effectiveness of virtual reality in the rehabilitation of motor function of patients with subacute stroke: A meta-analysis," *Frontiers in Neurology*, vol. 12, article 639535, 2021, doi: 10.3389/fneur.2021.639535.

28. D. B. Mekbib, J. Han, L. Zhang, et al., "Virtual reality therapy for upper limb rehabilitation in patients with stroke: a meta-analysis of randomized clinical trials," *Brain Injury*, vol. 34, no. 4, pp. 456-465, 2020, doi: 10.1080/02699052.2020.1725126.

29. S. Haeyen, S. van Hooren, W. M. van der Veld, et al., "Efficacy of art therapy in individuals with personality disorders cluster B/C" *Journal of Personality Disorders*, vol. 32, no. 4, pp. 527-542, 2018, doi: 10.1521/pedi\_2017\_31\_312.

30. G. Kaimal, K. Ray, and J. Muniz, "Reduction of cortisol levels and participants responses following art making" *Art Therapy*, vol. 33, no. 2, pp. 74-80, 2016, doi: 10.1080/07421656.2016.1166832.

31. T. L. Lo, J. L. Lee, and R. T. H. Ho, "Creative arts-based therapies for stroke survivors: A qualitative systematic review" *Frontiers in Psychology*, vol. 9, article 1646, 2018, doi: 10.3389/fpsyg.2018.01646.

32. R. Kongkasuwan, K. Voraakhom, P. Pisolayabutra, et al., "Creative art therapy to enhance rehabilitation for stroke patients" *Clinical Rehabilitation*, vol. 30, no. 10, pp. 1016-1023, 2016, doi: 10.1177/0269215515607072.

33. R. K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175-184, 1970.

34. G. Kwakkel, E. E. H. van Wegen, J. H. Burrridge, et al., "Standardized measurement of quality of upper limb movement after stroke: Consensus-based core recommendations from the Second Stroke Recovery and Rehabilitation Roundtable," *Neurorehabilitation and Neural Repair*, vol. 33, no. 11, pp. 951-958, 2019, doi: 10.1177/1545968319886477.

35. B. Lim, and S. Zohren, "Time-series forecasting with deep learning: a survey" *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, article 20200209, 2021, doi: 10.1098/rsta.2020.0209.

36. Q. Li, L. Chi, W. Zhao, L. Wu, C. Jiao, X. Zheng, K. Zhang, and X. Li, "Machine learning prediction of motor function in chronic stroke patients: a systematic review and meta-analysis," *Frontiers in Neurology*, vol. 14, article 1039794, 2023, doi: 10.3389/fneur.2023.1039794.

37. J. Liang, Z. Shi, F. Zhu, W. Chen, X. Chen, and Y. Li, "Gaussian process autoregression for joint angle prediction based on sEMG signals," *Frontiers in Public Health*, vol. 9, article 685596, 2021, doi: 10.3389/fpubh.2021.685596.

38. M. Dyvak, A. Melnyk, "Ontology of mathematical modeling based on interval data" *Complexity*, vol. 2022, article 8062969, 2022, doi: 10.1155/2022/8062969.

39. D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm" *Journal of Global Optimization*, vol. 39, no. 3, pp. 459-471, 2007, doi: 10.1007/s10898-007-9149-x.

40. Ю. В. Кіндзерський, "Методи та засоби підвищення ефективності паралельних обчислень на графічних процесорах для задач оптимізації" дис. ... д-ра філос.: 121, Тернопіль, 2026. 200 с.

41. R. T. Sutton, D. Pincock, D. C. Baumgart, et al., "An overview of clinical decision support systems: benefits, risks, and strategies for success" *npj Digital Medicine*, vol. 3, article 33, 2020, doi: 10.1038/s41746-020-0221-y.

42. E. H. Shortliffe, and M. J. Sepulveda, "Clinical decision support in the era of artificial intelligence" *JAMA*, vol. 320, no. 21, pp. 2199-2200, 2018, doi: 10.1001/jama.2018.17163.

43. WE/AR Studio. Vidnova Mixed Reality App – Case Study. 2024. URL: <https://wear-studio.com/case-item/vidnova/> (дата звернення: 03.06.2026).

44. G. Tieri, M. Iosa, A. Fortini, et al., "Efficacy of a virtual reality rehabilitation protocol based on art therapy in patients with stroke: A single-blind randomized controlled trial" *Brain Sciences*, vol. 14, no. 9, article 863, 2024, doi: 10.3390/brainsci14090863.

45. R. Tykhyi, Ya. Tsapiv, M. Dyvak, D. Popovych, A. Havrylenko, and T. Husieva, "Software system for supporting art therapy processes using augmented reality technologies" in *Proc. 2024 IEEE International Conference on Advanced Computer and Information Technologies (ACIT)*, 2024, doi: 10.1109/ACIT62333.2024.10712584.

46. Y. Tsapiv, M. Dyvak, R. Tykhyi, D. Popovych, and A. Havrylenko, "Automated goniometric analysis system based on neural networks" in *Proc. 2025 15th*

International Conference on Advanced Computer Information Technologies (ACIT-2025), Ceske Budejovice, Czech Republic, 2025, pp. 1-5.

47. Я. А. Цапів, А. В. Пукас, "Автоматизована безмаркерна система вимірювання діапазонів рухів суглобів на основі трикамерного відеоаналізу," Наукові праці ДонНТУ. Серія "Інформатика, кібернетика та обчислювальна техніка", Дрогобич, 2026, No. 1 (42), с. 54-64. ISSN 1996-1588.

48. Y. Tsapiv, M. Dyvak, A. Pukas, Y. Petrovskyi, A. Melnyk, A. Dyvak, A. Banasik, A. Czupryna-Nowak, P. Pikiewicz, Y. Popyk, and Y. Dzyha, "AR Technology for Restoring Upper Limb Joint Mobility in Patients," Applied Sciences, vol. 17, 2026 (подано до друку). ISSN 2076-3417.

49. Я. А. Цапів, А. В. Пукас, Д. Біловус, "Рекомендаційна система підтримки фізичної реабілітації на основі корекції терапевтичних зображень," Вимірювальна та обчислювальна техніка в технологічних процесах, Хмельницький, 2026 (подано до друку). ISSN 2219-9365.

50. NestJS. A progressive Node.js framework. URL: <https://nestjs.com> (дата звернення: 03.06.2026).

51. Christ T., Guseva S., Engel T. et al. We must not neglect rehabilitation in Ukraine // eClinicalMedicine (The Lancet). - 2022. - Vol. 50. - 101528. - URL: [https://www.thelancet.com/journals/eclinm/article/PIIS2589-5370\(22\)00267-X/fulltext](https://www.thelancet.com/journals/eclinm/article/PIIS2589-5370(22)00267-X/fulltext) (дата звернення: 03.06.2026).

52. World Health Organization. Overcoming barriers to accessing rehabilitation in Ukraine amidst conflict. - Geneva: WHO Europe, 2023. - URL: <https://www.who.int/europe/news/item/05-04-2023-overcoming-barriers-to-accessing-rehabilitation-in-ukraine-amidst-conflict> (дата звернення: 03.06.2026).

53. J. Bernhardt, K. S. Hayward, N. Dancause, et al., "A stroke recovery trial development framework: Consensus-based core recommendations from the Second Stroke Recovery and Rehabilitation Roundtable," International Journal of Stroke, vol. 14, no. 8, pp. 792-802, 2019, doi: 10.1177/1747493019879657.

54. K. E. Laver, B. Lange, S. George, J. E. Deutsch, G. Saposnik, M. Chapman, and M. Crotty, "Virtual reality for stroke rehabilitation" *Cochrane Database of Systematic Reviews*, issue 6, article CD008349, 2025, doi: 10.1002/14651858.CD008349.pub5.

55. M. Soleimani, M. Ghazisaeedi, and S. Heydari, "The efficacy of virtual reality for upper limb rehabilitation in stroke patients: a systematic review and meta-analysis" *BMC Medical Informatics and Decision Making*, vol. 24, article 139, 2024, doi: 10.1186/s12911-024-02534-y.

56. I. Bortone, D. Leonardis, M. Solazzi, et al., "Wearable haptics and immersive virtual reality rehabilitation training in children with neuromotor impairments" *IEEE Trans. Neural Systems and Rehabilitation Engineering*, vol. 26, no. 7, pp. 1469-1478, 2018, doi: 10.1109/TNSRE.2018.2846814.

57. M. J. Vinolo Gil, G. Gonzalez-Medina, D. Lucena-Anton, V. Perez-Cabezas, M. C. Ruiz-Molinero, and R. Martín-Valero, "Augmented reality in physical therapy: Systematic review and meta-analysis" *JMIR Serious Games*, vol. 9, no. 4, article e30985, 2021, doi: 10.2196/30985.

58. Z. Zhang, et al., "Effects of virtual reality with different modalities on upper limb recovery: A systematic review and network meta-analysis on optimizing stroke rehabilitation" *Frontiers in Neurology*, vol. 16, article 1544135, 2025, doi: 10.3389/fneur.2025.1544135.

59. J. H. Lee, J. H. Kim, S. H. Kim, et al., "The effects of virtual reality training on function in chronic stroke patients: A systematic review and meta-analysis" *BioMed Research International*, vol. 2019, article 7595639, 2019, doi: 10.1155/2019/7595639.

60. A. Elor, and S. Kurniawan, "The ultimate display for physical rehabilitation: A bridging review on immersive virtual reality" *Frontiers in Virtual Reality*, vol. 1, article 585993, 2020, doi: 10.3389/frvir.2020.585993.

61. J. J. Sánchez-Gil, A. Sáez-Manzano, R. López-Luque, J.-J. Ochoa-Sepúlveda, and E. Cañete-Carmona, "Gamified devices for stroke rehabilitation: A systematic review" *Computer Methods and Programs in Biomedicine*, vol. 258, article 108476, 2025, doi: 10.1016/j.cmpb.2024.108476.

62. S. M. N. Glegg, and D. E. Levac, "Barriers, facilitators and interventions to support virtual reality implementation in rehabilitation: A scoping review" *PM&R*, vol. 10, no. 11, pp. 1237-1251, 2018, doi: 10.1016/j.pmrj.2018.07.004.

63. G. Gebreheat, A. Goman, and A. Porter-Armstrong, "The use of home-based digital technology to support post-stroke upper limb rehabilitation: A scoping review" *Clinical Rehabilitation*, vol. 38, no. 1, pp. 60-71, 2024, doi: 10.1177/02692155231189257.

64. R. Joschko, C. Klatte, W. A. Grabowska, S. Roll, A. Berghöfer, and S. N. Willich, "Active visual art therapy and health outcomes: A systematic review and meta-analysis" *JAMA Network Open*, vol. 7, no. 9, article e2428709, 2024, doi: 10.1001/jamanetworkopen.2024.28709.

65. E. L. Brown, M. E. Gannotti, and D. A. Veneri, "Including arts in rehabilitation enhances outcomes in the psychomotor, cognitive, and affective domains: A scoping review" *Physical Therapy*, vol. 102, no. 4, article pzac003, 2022, doi: 10.1093/ptj/pzac003.

66. A. Christiansen, M. Scythes, B. R. Ritsma, S. H. Scott, and V. DePaul, "Art skill-based rehabilitation training for upper limb sensorimotor recovery post-stroke: A feasibility study" *Clinical Rehabilitation*, vol. 36, no. 10, pp. 1372-1384, 2022, doi: 10.1177/02692155221105586.

67. M. Iosa, M. Aydin, C. Candelise, et al., "The Michelangelo effect: Art improves the performance in a virtual reality task developed for upper limb neurorehabilitation" *Frontiers in Psychology*, vol. 11, article 611956, 2021, doi: 10.3389/fpsyg.2020.611956.

68. J. Tan, L. Kannis-Dymand, and C. Jones, "Examining the potential of VR program Tilt Brush in reducing anxiety," *Virtual Reality*, vol. 27, pp. 3379-3391, 2023, doi: 10.1007/s10055-022-00711-w.

69. C. Hadjipanayi, D. Banakou, and D. Michael-Grigoriou, "Art as therapy in virtual reality: A scoping review," *Frontiers in Virtual Reality*, vol. 4, article 1065863, 2023, doi: 10.3389/frvir.2023.1065863.

70. F. Shojaei, F. Shojaei, J. Osorio Torres, and P. C. Shih, "Insights from art therapists on using AI-generated art in art therapy: Mixed methods study" *JMIR Formative Research*, vol. 8, article e63038, 2024, doi: 10.2196/63038.

71. Magic Leap. Magic Leap 2 Developer Documentation. URL: <https://developer-docs.magicleap.cloud> (дата звернення: 03.06.2026).

72. G. E. Hancock, T. Herworth, and K. Wembridge, "Accuracy and reliability of knee goniometry methods" *Journal of Experimental Orthopaedics*, vol. 5, article 46, 2018, doi: 10.1186/s40634-018-0161-5.

73. P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A study of Vicon system positioning performance" *Sensors*, vol. 17, no. 7, article 1591, 2017, doi: 10.3390/s17071591.

74. A. Tharatipyakul, T. Srikaewsiew, and S. Pongnumkul, "Deep learning-based human body pose estimation in providing feedback for physical movement: A review," *Heliyon*, vol. 10, no. 17, article e36589, 2024, doi: 10.1016/j.heliyon.2024.e36589.

75. W. W. T. Lam, Y. M. Tang, and K. N. K. Fong, "A systematic review of the applications of markerless motion capture technology for clinical measurement in rehabilitation" *Journal of NeuroEngineering and Rehabilitation*, vol. 20, article 57, 2023, doi: 10.1186/s12984-023-01186-9.

76. L. Needham, M. Evans, D. P. Cosker, et al., "The accuracy of several pose estimation methods for 3D joint centre localisation" *Scientific Reports*, vol. 11, article 20673, 2021, doi: 10.1038/s41598-021-00212-x.

77. F. Roggio, B. Trovato, M. Sortino, and G. Musumeci, "A comprehensive analysis of the machine learning pose estimation models used in human movement and posture analyses: A narrative review," *Heliyon*, vol. 10, no. 21, article e39977, 2024, doi: 10.1016/j.heliyon.2024.e39977.

78. M. Slembrouck, H. Luong, J. Gerlo, et al., "Multiview 3D markerless human pose estimation from OpenPose skeletons," in *Proc. Advanced Concepts for Intelligent Vision Systems*, 2020, pp. 166-178.

79. Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose++: Vision transformer for generic body pose estimation" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 2, pp. 1185-1200, 2024, doi: 10.1109/TPAMI.2023.3330016.

80. C. Dong, and G. Du, "An enhanced real-time human pose estimation method based on modified YOLOv8 framework" *Scientific Reports*, vol. 14, article 8012, 2024, doi: 10.1038/s41598-024-58146-z.

81. Intel Corporation, "OpenVINO™ Toolkit – Open Model Optimizer and Inference Engine" 2024. [Online]. Available: <https://docs.openvino.ai/>. (Accessed: 03.06.2026).

82. M. D'Haene, F. Chorin, S. S. Colson, O. Guérin, R. Zory, and E. Piche, "Validation of a 3D markerless motion capture tool using multiple pose and depth estimations for quantitative gait analysis" *Sensors*, vol. 24, no. 22, article 7105, 2024, doi: 10.3390/s24227105.

83. W. W. T. Lam, and K. N. K. Fong, "Validity and reliability of upper limb kinematic assessment using a markerless motion capture (MMC) system: A pilot study," *Archives of Physical Medicine and Rehabilitation*, vol. 105, pp. 673-681, 2024, doi: 10.1016/j.apmr.2023.10.018.

84. O. Pottorf, D. Vapne, J. Ghigiarelli, and K. Haase, "Reliability and concurrent validity of a markerless, single camera, portable 3D motion capture system for assessment of glenohumeral mobility" *International Journal of Sports Physical Therapy*, vol. 18, no. 5, pp. 1176-1185, 2023, doi: 10.26603/001c.88003.

85. L. C. Klein, A. A. Chellal, V. Grilo, J. Goncalves, M. F. Pacheco, F. P. Fernandes, F. C. Monteiro, and J. Lima, "Assessing the reliability of AI-based angle detection for shoulder and elbow rehabilitation," in *Proc. Optimization, Learning Algorithms and Applications*, Springer, 2024, pp. 3-18, doi: 10.1007/978-3-031-53036-4\_1.

86. L. F. Yeung, Z. Yang, K. C. C. Cheng, et al., "Effects of camera viewing angles on tracking kinematic gait patterns using Azure Kinect, Kinect v2 and Orbbec Astra Pro v2" *Gait & Posture*, vol. 87, pp. 19-26, 2021, doi: 10.1016/j.gaitpost.2021.04.005.

87. J. A. Albert, V. Owolabi, A. Gebel, et al., "Evaluation of the pose tracking performance of the Azure Kinect and Kinect v2 for gait analysis in comparison with a gold standard: A pilot study" *Sensors*, vol. 20, no. 18, article 5104, 2020, doi: 10.3390/s20185104.

88. R. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge: Cambridge University Press, 2004. ISBN: 978-0-521-54051-3.

89. A. F. R. Nogueira, H. P. Oliveira, and L. F. Teixeira, "Markerless multi-view 3D human pose estimation: A survey," *Image and Vision Computing*, vol. 155, article 105437, 2025, doi: 10.1016/j.imavis.2025.105437.

90. D. Karaboga, and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," in *Proc. IFSA 2007*, LNCS, vol. 4529, pp. 789-798, Springer, 2007, doi: 10.1007/978-3-540-72950-1\_77.

91. GBD 2021 Stroke Collaborators, "Global, regional, and national burden of stroke and its risk factors, 1990–2021: a systematic analysis for the Global Burden of Disease Study 2021" *The Lancet Neurology*, vol. 23, no. 10, pp. 973-1003, 2024, doi: 10.1016/S1474-4422(24)00369-7.

92. L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999. ISBN: 978-0-13-656695-3.

93. A. Patharkar, F. Cai, F. Al-Hindawi, and T. Wu, "Predictive modeling of biomedical temporal data in healthcare applications: review and future directions," *Frontiers in Physiology*, vol. 15, article 1386760, 2024, doi: 10.3389/fphys.2024.1386760.

94. C. M. Stinear, M.-C. Smith, and W. D. Byblow, "Prediction tools for stroke rehabilitation," *Stroke*, vol. 50, pp. 3314-3322, 2019, doi: 10.1161/STROKEAHA.119.025696.

95. S. Seoni, V. Jahmunah, M. Salvi, P. D. Barua, F. Molinari, and U. R. Acharya, "Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023)" *Computers in Biology and Medicine*, vol. 165, article 107441, 2023, doi: 10.1016/j.compbimed.2023.107441.

96. R. E. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

97. L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. London: Springer-Verlag, 2001. ISBN: 978-1-4471-0249-6, doi: 10.1007/978-1-4471-0249-6.
98. X.-S. Yang, *Nature-Inspired Optimization Algorithms*. London: Elsevier, 2014. ISBN: 978-0-12-416743-8, doi: 10.1016/C2013-0-01368-0.
99. J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975. ISBN: 978-0-262-58111-0.
100. J. Kennedy, and R. Eberhart, "Particle swarm optimization" in *Proc. IEEE International Conference on Neural Networks (ICNN)*, vol. 4, Perth, Australia, 1995, pp. 1942-1948, doi: 10.1109/ICNN.1995.488968.
101. M. Dorigo, and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004. ISBN: 978-0-262-04219-2.
102. D. Karaboga, "An idea based on honey bee swarm for numerical optimization" Technical Report TR06, Erciyes University, Department of Computer Engineering, 2005.
103. M. Dyvak, I. Spivak, A. Melnyk, V. Manzhula, T. Dyvak, A. Rot, and M. Hernes, "Modeling based on the analysis of interval data of atmospheric air pollution processes with nitrogen dioxide" *Sustainability*, vol. 15, article 2163, 2023, doi: 10.3390/su15032163.
104. М. П. Дивак, Н. П. Порплиця, та О. В. Мацюк, "Ідентифікація моделей профілю користувача соціальних мереж на основі інтервальних даних" *Індуктивне моделювання складних систем*, вип. 10, с. 56-66, 2018.
105. D. Karaboga, B. Akay, and N. Karaboga, "A survey on the studies employing machine learning (ML) for enhancing artificial bee colony (ABC) optimization algorithm," *Cogent Engineering*, vol. 7, no. 1, article 1855741, 2020, doi: 10.1080/23311916.2020.1855741.
106. G. Welch, and G. Bishop, "An introduction to the Kalman filter" Technical Report TR 95-041, Chapel Hill: University of North Carolina, 1995. URL: [https://www.cs.unc.edu/~welch/media/pdf/kalman\\_intro.pdf](https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf) (дата звернення: 03.06.2026).

107. M. J. Kolber, F. Vega, K. Widmayer, and M.-S. S. Cheng, "The reliability and minimal detectable change of shoulder mobility measurements using a digital inclinometer" *Physiotherapy Theory and Practice*, vol. 27, no. 2, pp. 176-184, 2011, doi: 10.3109/09593985.2010.481011.

108. M. Dyvak, O. Kindzersky, L. Dostalek, M. Stetsko, and J. Nowak, "Parallel computations in the problem of identification of interval discrete models based on swarm intelligence of a bee colony" in *Proc. 13th International Conference on Advanced Computer Information Technologies (ACIT)*, Wroclaw, Poland, 2023, pp. 23-28, doi: 10.1109/ACIT58437.2023.10275695.

109. M. Dyvak, and O. Kindzersky, "Implementation of the structural identification for interval models based on the behavioral model of an artificial bee colony" in *Proc. 15th International Conference on Advanced Computer Information Technologies (ACIT)*, Šibenik, Croatia, 2025, pp. 98-101, doi: 10.1109/ACIT65614.2025.11185828.

110. D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21-57, 2014, doi: 10.1007/s10462-012-9328-0.

111. F. Lanotte, M. K. O'Brien, and A. Jayaraman, "AI in rehabilitation medicine: Opportunities and challenges" *Annals of Rehabilitation Medicine*, vol. 47, no. 6, pp. 444-458, 2023, doi: 10.5535/arm.23131.

112. S. Kierner, J. Kucharski, and Z. Kierner, "Taxonomy of hybrid architectures involving rule-based reasoning and machine learning in clinical decision systems: A scoping review," *Journal of Biomedical Informatics*, vol. 144, article 104428, doi: 10.1016/j.jbi.2023.104428.

113. A. D. Pelosi, N. Roth, T. Yehoshua, et al., "Personalized rehabilitation approach for reaching movement using reinforcement learning," *Scientific Reports*, vol. 14, article 17675, 2024, doi: 10.1038/s41598-024-64514-6.

114. F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Techniques, applications, and challenges," in *Recommender Systems Handbook*, 3rd ed., New York: Springer, 2022, pp. 1–35, doi: 10.1007/978-1-0716-2197-4\_1.

115. M. A. Mehrabadi, E. Khatibi, T. Jimah, S. Labbaf, H. Borg, L. Narvaez, P. Pimentel, A. Turner, N. D. Dutt, Y. Guo, and A. M. Rahmani, "PERFECT: Personalized Exercise Recommendation Framework and architECTure" *ACM Transactions on Computing for Healthcare*, vol. 5, no. 4, 2024, doi: 10.1145/3696425.
116. A. Esteva, A. Robicquet, B. Ramsundar, et al., "A guide to deep learning in healthcare" *Nature Medicine*, vol. 25, pp. 24-29, 2019, doi: 10.1038/s41591-018-0316-z.
117. A. Abedi, T. J. F. Colella, M. Pakosh, and D. Bhatt, "Artificial intelligence-driven virtual rehabilitation for people living in the community: A scoping review" *npj Digital Medicine*, vol. 7, article 25, 2024, doi: 10.1038/s41746-024-00998-w.
118. H. Wang, J. Guo, Y. Zhang, Z. Fu, and Y. Yao, "Closed-loop rehabilitation of upper-limb dyskinesia after stroke: from natural motion to neuronal microfluidics" *Journal of NeuroEngineering and Rehabilitation*, vol. 22, article 87, 2025, doi: 10.1186/s12984-025-01617-9.
119. A. Zubala, A. Pease, K. Lyszkiewicz, and S. Hackett, "Art psychotherapy meets creative AI: An integrative review positioning the role of creative AI in art therapy process" *Frontiers in Psychology*, vol. 16, article 1548396, 2025, doi: 10.3389/fpsyg.2025.1548396.
120. A. R. Rasa, "Artificial intelligence and its revolutionary role in physical and mental rehabilitation: A review of recent advancements" *BioMed Research International*, vol. 2024, article 9554590, 2024, doi: 10.1155/bmri/9554590.
121. S. Y. Kim, D. H. Kim, M. J. Kim, H. J. Ko, and O. R. Jeong, "XAI-based clinical decision support systems: A systematic review" *Applied Sciences*, vol. 14, no. 15, article 6638, 2024, doi: 10.3390/app14156638.
122. A. H. Mohamed, and K. P. Schwarz, "Adaptive Kalman filtering for INS/GPS," *Journal of Geodesy*, vol. 73, pp. 193-203, 1999, doi: 10.1007/s001900050236.
123. M. Moro, G. L. Marchesi, F. Hichri, L. Casini, and S. Uras, "Assessment of monocular human pose estimation models for clinical movement analysis," *Scientific Reports*, vol. 15, article 2849, 2025, doi: 10.1038/s41598-025-22626-7.
124. American Academy of Orthopaedic Surgeons, *Joint Motion: Method of Measuring and Recording*. Chicago: AAOS, 1994.

125. N. Ocheretnyuk, I. Voytyuk, M. Dyvak, and Y. Martsenyuk, "Features of structure identification the macromodels for nonstationary fields of air pollutions from vehicles" in Proc. 11th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science, Lviv, Ukraine, 2012, p. 444.

126. M. Dyvak, and N. Porplytsya, "Parameters identification method of interval discrete dynamic models of air pollution based on artificial bee colony algorithm" in Proc. ACIT, 2020, pp. 130-135.

127. M. Dyvak, N. Porplytsya, Y. Maslyiak, and N. Kasatkina, "Modified artificial bee colony algorithm for structure identification of models of objects with distributed parameters and control" in Proc. CADSM, 2017, pp. 50-54.

128. D. Karaboga, and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, vol. 8, no. 1, pp. 687-697, 2008, doi: 10.1016/j.asoc.2007.05.007.

129. I. C. Brunner, E.-R. Andrinopoulou, R. Selles, and C. B. L. Lundquist, "External validation of a dynamic prediction model for upper limb function after stroke" Archives of Rehabilitation Research and Clinical Translation, vol. 5, article 100315, 2024, doi: 10.1016/j.arrct.2023.100315.

130. H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)" RFC 2326, IETF, 1998.

131. Google LLC. MediaPipe Solutions Guide - Pose Landmarker. URL: [https://developers.google.com/mediapipe/solutions/vision/pose\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/pose_landmarker) (дата звернення: 03.06.2026).

132. Unity Technologies. OpenXR Plugin Documentation. URL: <https://docs.unity3d.com/Packages/com.unity.xr.openxr> (дата звернення: 03.06.2026).



```

        OEF_MINCUTOFF,
        OEF_BETA,
        OEF_DCUTOFF,
        MIN_KP_CONF,
    )
    except Exception:
        self._native = None

def smooth(self, kps: np.ndarray) -> np.ndarray:
    """
    kps: (N, 3) float32 - x, y, confidence.
    Returns smoothed array of the same shape. Points with confidence
    below MIN_KP_CONF are passed through unchanged so that low-quality
    detections do not poison the filter state.
    """
    if self._native is not None:
        return np.asarray(self._native.smooth(kps), dtype=np.float32)

    out = kps.copy().astype(np.float32)
    for i, (fx, fy) in enumerate(self._f):
        if i >= len(kps):
            break
        if kps[i, 2] >= MIN_KP_CONF:
            out[i, 0] = fx(float(kps[i, 0]))
            out[i, 1] = fy(float(kps[i, 1]))
    return out

def reset(self):
    if self._native is not None:
        self._native.reset()
    return
    for pair in self._f:
        for f in pair:
            f.reset()

# =====
# GEOMETRIC PRIMITIVES
# =====

def angle_3pts(a: np.ndarray, b: np.ndarray, c: np.ndarray) -> float:
    """
    Angle in degrees at vertex b formed by points a-b-c.
    Returns a value in [0, 180].
    """
    ba = a - b
    bc = c - b
    cos = np.dot(ba, bc) / (np.linalg.norm(ba) * np.linalg.norm(bc) + 1e-9)
    return float(np.degrees(np.arccos(np.clip(cos, -1.0, 1.0))))

def signed_angle_2d(v1: np.ndarray, v2: np.ndarray) -> float:
    """Signed angle in degrees between two 2D vectors, range [-180, 180]."""
    a = math.atan2(float(np.cross(v1, v2)), float(np.dot(v1, v2)))
    return math.degrees(a)

def midpoint(a: np.ndarray, b: np.ndarray) -> np.ndarray:
    return (a + b) / 2.0

def kp(pts: np.ndarray, idx: int) -> Optional[np.ndarray]:
    """Return (x, y) of keypoint idx, or None if confidence is too low."""
    if idx >= len(pts) or pts[idx, 2] < MIN_KP_CONF:
        return None
    return pts[idx, :2].astype(np.float32)

# =====

```

```

# BEST CAMERA SELECTION FOR A JOINT
# =====

class CameraSelector:
    """
    For each anatomical joint, pick the camera whose keypoints used in
    the angle computation have the highest mean confidence. Ties are
    resolved using a static priority list from config.JOINT_BEST_VIEW
    so that anatomically preferable views (e.g., side cameras for elbow
    flexion) win when confidences are equal.
    """

    def best(
        self,
        joint_name: str,
        kp_indices: List[int],
        cam_frames: Dict[str, Dict],
    ) -> Optional[str]:
        """
        joint_name : key into JOINT_BEST_VIEW.
        kp_indices : indices of YOLO-pose keypoints used for the angle.
        cam_frames : {cam_id: {"kps": (17,3), "hand_left": ..., "hand_right": ...}}.

        Returns the cam_id of the best camera or None.
        """
        priority = JOINT_BEST_VIEW.get(joint_name, list(cam_frames.keys()))
        best_cam = None
        best_conf = -1.0

        for cam_id, data in cam_frames.items():
            pts = data.get("kps")
            if pts is None:
                continue
            confs = [
                float(pts[i, 2]) if i < len(pts) else 0.0
                for i in kp_indices
            ]
            score = sum(confs) / len(confs) if confs else 0.0

            # Small tie-breaker bonus from the static priority list.
            try:
                pri_bonus = 0.01 * (len(priority) - priority.index(cam_id))
            except ValueError:
                pri_bonus = 0.0

            total = score + pri_bonus
            if total > best_conf:
                best_conf = total
                best_cam = cam_id

        return best_cam

# =====
# JOINT ANGLE COMPUTATION
# =====

# YOLO-pose indices (COCO 17 kp):
# 0 nose      1 left_eye    2 right_eye    3 left_ear    4 right_ear
# 5 left_shoulder  6 right_shoulder
# 7 left_elbow    8 right_elbow
# 9 left_wrist    10 right_wrist
# 11 left_hip     12 right_hip
# 13 left_knee    14 right_knee
# 15 left_ankle   16 right_ankle

L_SHOULDER, R_SHOULDER = 5, 6
L_ELLOW, R_ELLOW = 7, 8
L_WRIST, R_WRIST = 9, 10
L_HIP, R_HIP = 11, 12
L_KNEE, R_KNEE = 13, 14

```

```

L_ANKLE,   R_ANKLE   = 15, 16
NOSE       = 0

# MediaPipe Hands indices (21 kp):
# 0 wrist  4 thumb tip  5 index MCP  8 index tip
# 9 middle MCP  13 ring MCP  17 pinky MCP  ...
MP_WRIST = 0
MP_INDEX = 5
MP_PINKY = 17
MP_THUMB = 4

class AngleResult:
    """
    Container for a single computed joint angle and its clinical status.
    The status is derived from the value, the joint's normal range, and
    a tolerance margin (ANGLE_WARN_MARGIN, in degrees).
    """
    __slots__ = ("name", "value", "normal_range", "status", "source_cam")

    def __init__(self, name: str, value: float, normal_range: Tuple[float, float], source_cam:
str):
        self.name          = name
        self.value         = value
        self.normal_range = normal_range
        self.source_cam    = source_cam

        lo, hi = normal_range
        margin = ANGLE_WARN_MARGIN
        if lo - margin <= value <= hi + margin:
            self.status = "ok" if lo <= value <= hi else "warn"
        else:
            self.status = "bad"

class AngleEngine:
    """
    Computes all clinically relevant angles for the shoulder, elbow,
    and wrist joints of both arms. Aggregates data from all cameras
    and selects the best view per joint.
    """

    def __init__(self):
        self.selector = CameraSelector()
        self._native = None
        if _pose_native is not None:
            try:
                self._native = _pose_native.NativeAngleEngine(
                    MIN_KP_CONF,
                    ANGLE_WARN_MARGIN,
                    JOINT_RANGES,
                    JOINT_BEST_VIEW,
                )
            except Exception:
                self._native = None

# ---- low-level helpers -----

def _get(self, cam_frames: Dict, cam_id: str, idx: int) -> Optional[np.ndarray]:
    """Return YOLO keypoint (x, y) from a given camera, or None."""
    d = cam_frames.get(cam_id)
    if d is None:
        return None
    return kp(d["kps"], idx)

def _get_hand(self, cam_frames: Dict, cam_id: str, idx: int,
side: str) -> Optional[np.ndarray]:
    """Return MediaPipe hand keypoint (x, y) for left or right hand."""
    d = cam_frames.get(cam_id)
    if d is None:
        return None

```

```

hand = d.get(f"hand_{side}")
if hand is None:
    return None
return kp(hand, idx)

def _angle_or_none(
    self, a: Optional[np.ndarray],
    b: Optional[np.ndarray],
    c: Optional[np.ndarray],
) -> Optional[float]:
    if a is None or b is None or c is None:
        return None
    return angle_3pts(a, b, c)

def _result(self, name: str, value: Optional[float],
            cam_id: str) -> Optional[AngleResult]:
    if value is None:
        return None
    rang = JOINT_RANGES.get(name, (0, 180))
    return AngleResult(name, value, rang, cam_id)

# ---- shoulder joint -----

def _shoulder_angles(
    self, cam_frames: Dict, side: str
) -> List[Optional[AngleResult]]:
    """
    Compute up to three angles for one shoulder joint.
    side: "left" | "right".
    """
    S = L_SHOULDER if side == "left" else R_SHOULDER
    E = L_ELBOW   if side == "left" else R_ELBOW
    H = L_HIP     if side == "left" else R_HIP
    oS = R_SHOULDER if side == "left" else L_SHOULDER # opposite shoulder

    results = []
    sfx = " L" if side == "left" else " R"

    # 1. Flexion / extension (sagittal plane -> front camera).
    cam1 = self.selector.best("Плече згинання", [S, E, H], cam_frames)
    if cam1:
        sh = self._get(cam_frames, cam1, S)
        el = self._get(cam_frames, cam1, E)
        hi = self._get(cam_frames, cam1, H)
        ang = self._angle_or_none(hi, sh, el)
        results.append(self._result("Плече згинання" + sfx, ang, cam1))

    # 2. Abduction (frontal plane -> front or contralateral camera).
    cam2 = self.selector.best("Плече абдукція", [S, E, oS], cam_frames)
    if cam2:
        sh = self._get(cam_frames, cam2, S)
        el = self._get(cam_frames, cam2, E)
        osh = self._get(cam_frames, cam2, oS)
        ang = self._angle_or_none(osh, sh, el)
        results.append(self._result("Плече абдукція" + sfx, ang, cam2))

    # 3. Horizontal flexion/extension. The transverse plane is
    # approximated using a side camera and the angle between the
    # arm vector and the inter-shoulder line.
    cam3 = self.selector.best("Плече гориз. згинання", [S, E], cam_frames)
    if cam3:
        sh = self._get(cam_frames, cam3, S)
        el = self._get(cam_frames, cam3, E)
        os = self._get(cam_frames, cam3, oS)
        if sh is not None and el is not None and os is not None:
            arm_vec = el - sh
            trunk_vec = os - sh
            ang = abs(signed_angle_2d(trunk_vec, arm_vec))
            results.append(self._result("Плече гориз. згинання" + sfx, ang, cam3))

    return [r for r in results if r is not None]

```

```

# ---- elbow joint -----

def _elbow_angles(
    self, cam_frames: Dict, side: str
) -> List[Optional[AngleResult]]:
    S = L_SHOULDER if side == "left" else R_SHOULDER
    E = L_ELBOW   if side == "left" else R_ELBOW
    W = L_WRIST   if side == "left" else R_WRIST
    sfx = " L" if side == "left" else " R"

    results = []

    # 1. Elbow flexion (3-point angle: shoulder-elbow-wrist).
    cam = self.selector.best("Лікоть згинання", [S, E, W], cam_frames)
    if cam:
        sh = self._get(cam_frames, cam, S)
        el = self._get(cam_frames, cam, E)
        wr = self._get(cam_frames, cam, W)
        ang = self._angle_or_none(sh, el, wr)
        results.append(self._result("Лікоть згинання" + sfx, ang, cam))

    # 2. Pronation / supination - coarse approximation via the
    # forearm orientation relative to the horizontal axis.
    cam2 = self.selector.best("Лікоть пронація", [E, W], cam_frames)
    if cam2:
        el = self._get(cam_frames, cam2, E)
        wr = self._get(cam_frames, cam2, W)
        if el is not None and wr is not None:
            v = wr - el
            ang = abs(math.degrees(math.atan2(float(v[1]), float(v[0]))))
            results.append(self._result("Лікоть пронація" + sfx, ang, cam2))

    return [r for r in results if r is not None]

# ---- wrist joint -----

def _wrist_angles(
    self, cam_frames: Dict, side: str
) -> List[Optional[AngleResult]]:
    E = L_ELBOW if side == "left" else R_ELBOW
    W = L_WRIST if side == "left" else R_WRIST
    sfx = " L" if side == "left" else " R"
    results = []

    # Forearm vector (elbow -> wrist) comes from YOLO; hand axis
    # vector (wrist -> index MCP) comes from MediaPipe Hands.
    cam = self.selector.best("Зап'ястя долонне", [E, W], cam_frames)
    if cam:
        el = self._get(cam_frames, cam, E)
        wr = self._get(cam_frames, cam, W)
        mcp = self._get_hand(cam_frames, cam, MP_INDEX, side)

        if el is not None and wr is not None and mcp is not None:
            forearm = wr - el
            hand = mcp - wr
            ang = angle_3pts(el, wr, mcp)

            # Distinguish palmar (flexion) from dorsal (extension)
            # by the sign of the cross product.
            cross = np.cross(forearm, hand)
            if cross > 0:
                results.append(self._result("Зап'ястя долонне" + sfx, ang, cam))
            else:
                results.append(self._result("Зап'ястя тильне" + sfx, ang, cam))

    # Radial / ulnar deviation (abduction / adduction).
    cam2 = self.selector.best("Зап'ястя абдукція", [E, W], cam_frames)
    if cam2:
        el = self._get(cam_frames, cam2, E)
        wr = self._get(cam_frames, cam2, W)
        idx_kp = self._get_hand(cam_frames, cam2, MP_INDEX, side)

```

```

    pnk_kp = self._get_hand(cam_frames, cam2, MP_PINKY, side)

    if all(v is not None for v in [el, wr, idx_kp, pnk_kp]):
        hand_axis = idx_kp - pnk_kp
        forearm = wr - el
        ang = abs(signed_angle_2d(forearm, hand_axis))
        if ang > 90:
            ang = 180 - ang
        results.append(self._result("Зап'ястя абдукція" + sfx, ang, cam2))

    return [r for r in results if r is not None]

# ---- public entry point -----

def compute(self, cam_frames: Dict) -> List[AngleResult]:
    """
    cam_frames: {
        cam_id: {
            "kps": np.ndarray (17, 3), # YOLO-pose (x, y, conf)
            "hand_left": np.ndarray (21, 3) | None,
            "hand_right": np.ndarray (21, 3) | None,
        }
    }
    Returns a list of AngleResult (possibly empty).
    """
    if self._native is not None:
        try:
            native_items = self._native.compute(cam_frames)
            out: List[AngleResult] = []
            for item in native_items:
                nm = str(item["name"])
                val = float(item["value"])
                rng = tuple(item["normal_range"])
                sc = str(item["source_cam"])
                r = AngleResult(nm, val, (float(rng[0]), float(rng[1])), sc)
                st = item.get("status")
                if st in ("ok", "warn", "bad"):
                    r.status = st
                out.append(r)
            return out
        except Exception:
            self._native = None

    results: List[AngleResult] = []
    for side in ("left", "right"):
        results.extend(self._shoulder_angles(cam_frames, side))
        results.extend(self._elbow_angles(cam_frames, side))
        results.extend(self._wrist_angles(cam_frames, side))
    return results

```

## ДОДАТОК Б

# КОНВЕЄР БЕЗМАРКЕРНОГО ЗАХОПЛЕННЯ ТА НЕЙРОМЕРЕЖЕВОГО РЕЗУЛЬТАТУ

```

"""
pipeline.py - shared capture and inference pipeline (no Qt dependency).
Used by both the desktop UI and the FastAPI HTTP server.
"""

from __future__ import annotations

import json
import os
import shutil
import threading
import time
from datetime import datetime
from typing import Callable, Dict, List, Optional, Tuple

import cv2
import mediapipe as mp
import numpy as np
from ultralytics import YOLO

import config as cfg
from core import AngleEngine, KeypointSmoother, AngleResult
from renderer import Renderer

try:
    import pose_native as _pose_native # type: ignore
except Exception:
    _pose_native = None

# =====
# RTSP CAPTURE THREAD
# =====

class CaptureThread(threading.Thread):
    """
    Pulls frames from a single camera (RTSP, USB, or file) into a
    shared dictionary that the inference loop polls. Designed to be
    fully decoupled from the consumer so that capture FPS does not
    drop when inference temporarily slows down.
    """
    def __init__(self, cam_cfg: dict, shared: dict):
        super().__init__(daemon=True)
        self.cfg = cam_cfg
        self.shared = shared
        self._stop = threading.Event()

    def run(self):
        cam_id = self.cfg["id"]
        url = self.cfg["url"]
        print(f"[{cam_id}] Connecting to {url} ...")

        # Apply low-latency FFMPEG options before the first VideoCapture
        # call. This reduces RTSP startup buffering and the initial
        # I-frame burst cost on Tapo C200 cameras (and similar IP cams).
        rtsp_opts = getattr(cfg, "RTSP_FFMPEG_OPTIONS", "")
        is_rtsp = isinstance(url, str) and url.lower().startswith("rtsp://")
        if rtsp_opts and is_rtsp:
            os.environ["OPENCV_FFMPEG_CAPTURE_OPTIONS"] = rtsp_opts

        backend = cv2.CAP_FFMPEG if is_rtsp else cv2.CAP_ANY

        while not self._stop.is_set():

```

```

cap = cv2.VideoCapture(url, backend)
cap.set(cv2.CAP_PROP_BUFFERSIZE, cfg.RTSP_BUFFER)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, cfg.FRAME_W)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, cfg.FRAME_H)

if not cap.isOpened():
    print(f"[{cam_id}] Connection failed, retry in 3 s...")
    time.sleep(3)
    continue

print(f"[{cam_id}] Connected")
fail = 0

while not self._stop.is_set():
    ret, frame = cap.read()
    if not ret:
        fail += 1
        if fail > 30:
            print(f"[{cam_id}] Stream lost, reconnecting...")
            break
        time.sleep(0.03)
        continue

    fail = 0
    frame = cv2.resize(frame, (cfg.FRAME_W, cfg.FRAME_H))
    self.shared[cam_id] = {"frame": frame, "ts": time.time()}

cap.release()

def stop(self):
    self._stop.set()

# =====
# INFERENCE WORKER
# =====

class InferenceWorker:
    """
    Wraps YOLOv8-pose (body skeleton, 17 keypoints) and MediaPipe Hands
    (21 keypoints per hand) behind a single .process_batch entry point.
    Implements per-camera frame skipping so that pose detection and
    hand detection can run at independent cadences.
    """
    def __init__(self, yolo_batch_infer=True, hands_every_n_frames=1, pose_every_n_frames=1):
        self.yolo_batch_infer = bool(yolo_batch_infer)
        self.hands_every_n_frames = max(1, int(hands_every_n_frames))
        self.pose_every_n_frames = max(1, int(pose_every_n_frames))
        self.yolo, self.pose_backend = self._load_pose_model()
        self.yolo.overrides["verbose"] = False
        # ONNX/OpenVINO runtimes do not benefit from batched inference
        # the same way Torch does, so disable batching for them.
        if self.pose_backend in {"onnxruntime", "openvino"} and self.yolo_batch_infer:
            self.yolo_batch_infer = False
        print(f"Pose backend: {self.pose_backend}")

    # MediaPipe Hands has both a legacy and a modern API. We try the
    # legacy API first because it is more stable across versions.
    try:
        mp_hands = mp.solutions.hands
        self.hands = mp_hands.Hands(
            static_image_mode=False, max_num_hands=2,
            min_detection_confidence=0.5, min_tracking_confidence=0.45,
            model_complexity=0,
        )
        self._mp_legacy = True
    except AttributeError:
        from mediapipe.tasks import python as mp_tasks
        from mediapipe.tasks.python import vision as mp_vision
        import urllib.request
        model_path = "hand_landmarker.task"
        if not os.path.exists(model_path):

```

```

        print("Downloading hand_landmarker.task ...")
        urllib.request.urlretrieve(
            "https://storage.googleapis.com/mediapipe-models/hand_landmarker/"
            "hand_landmarker/float16/latest/hand_landmarker.task", model_path,
        )
        base_opts = mp_tasks.BaseOptions(model_asset_path=model_path)
        opts = mp_vision.HandLandmarkerOptions(
            base_options=base_opts, num_hands=2,
            min_hand_detection_confidence=0.5, min_hand_presence_confidence=0.45,
            min_tracking_confidence=0.45, running_mode=mp_vision.RunningMode.IMAGE,
        )
        self.hands = mp_vision.HandLandmarker.create_from_options(opts)
        self._mp_legacy = False

    # Per-camera state: smoothers, frame counters, last results.
    self.smoothers: Dict[str, KeypointSmoother] = {}
    self._hand_last: Dict[str, Tuple] = {}
    self._hand_tick: Dict[str, int] = {}
    self._pose_last: Dict[str, Optional[np.ndarray]] = {}
    self._pose_tick: Dict[str, int] = {}

    self._warmup()

def _warmup(self, max_iters: int = 30):
    """
    Force OpenVINO / ONNX Runtime to compile the execution graph
    before live capture starts. Runs inference on a dummy frame
    until per-pass latency stabilises (3 consecutive passes within
    15 % of each other) or max_iters is reached. This prevents
    JIT compilation from happening during the first session and
    halving FPS for the first 60-90 s.
    """
    if self.pose_backend not in {"openvino", "onnxruntime"}:
        return
    dummy = np.zeros((cfg.FRAME_H, cfg.FRAME_W, 3), dtype=np.uint8)
    print(f"[warmup] Pre-compiling {self.pose_backend} graph (up to {max_iters}
passes)...")
    times: List[float] = []
    for i in range(max_iters):
        t0 = time.perf_counter()
        self._run_yolo_single(dummy)
        times.append(time.perf_counter() - t0)
        if len(times) >= 4:
            last3 = times[-3:]
            if max(last3) <= min(last3) * 1.15:
                print(f"[warmup] Stable after {i + 1} passes - {times[-1] * 1000:.0f}
ms/frame")
                return
    print(f"[warmup] Done ({max_iters} passes) - {times[-1] * 1000:.0f} ms/frame")

def _load_pose_model(self):
    """
    Tries the configured pose backend in order: OpenVINO -> ONNX
    Runtime -> PyTorch. Each backend can auto-export from the .pt
    weights if its target file is missing. Falls back transparently
    on import or load errors so the application still starts on
    machines without OpenVINO / onnxruntime installed.
    """
    model_path = str(getattr(cfg, "YOLO_MODEL", "yolov8n-pose.pt"))
    runtime = str(getattr(cfg, "YOLO_RUNTIME", "torch")).lower()
    onnx_path = str(getattr(cfg, "YOLO_ONNX_MODEL", "yolov8n-pose.onnx"))
    ov_path = str(getattr(cfg, "YOLO_OPENVINO_MODEL", "yolov8n-pose_openvino_model"))
    ov_cache = str(getattr(cfg, "OPENVINO_CACHE_DIR", ".openvino_cache"))

    if runtime == "onnxruntime":
        try:
            import onnxruntime # noqa
            if not os.path.exists(onnx_path) and getattr(cfg, "YOLO_EXPORT_ONNX_IF_MISSING",
True):
                exporter = YOLO(model_path)
                exp = exporter.export(format="onnx", imgsz=cfg.YOLO_IMGSZ, opset=12,
simplify=True)
                if exp and os.path.exists(str(exp)) and os.path.abspath(str(exp)) !=
os.path.abspath(onnx_path):
                    shutil.copy2(str(exp), onnx_path)
                    if os.path.exists(onnx_path):

```

```

        return YOLO(onnx_path), "onnxruntime"
    except Exception as e:
        print(f"[warn] ONNX path failed ({e}), fallback to torch.")

if runtime == "openvino":
    try:
        import openvino as ov
        # Enable compiled-model disk cache so the graph is
        # never recompiled after the first run, saving 60-90 s
        # on every subsequent start.
        try:
            os.makedirs(ov_cache, exist_ok=True)
            core = ov.Core()
            core.set_property("CPU", {"CACHE_DIR": ov_cache})
            print(f"[openvino] Model cache enabled at '{ov_cache}'")
        except Exception as e:
            print(f"[openvino] Cache setup skipped: {e}")

        if not os.path.exists(ov_path) and getattr(cfg,
"YOLO_EXPORT_OPENVINO_IF_MISSING", True):
            YOLO(model_path).export(format="openvino", imgsz=cfg.YOLO_IMGSZ)
            if os.path.exists(ov_path):
                return YOLO(ov_path), "openvino"
    except Exception as e:
        print(f"[warn] OpenVINO path failed ({e}), fallback to torch.")

return YOLO(model_path), "torch"

def _ensure_smoother(self, cam_id):
    if cam_id not in self.smoothers:
        self.smoothers[cam_id] = KeypointSmoother(n_kp=17)

def _run_yolo_single(self, frame):
    """Single-frame YOLO inference. Returns (17, 3) array or None."""
    for r in self.yolo.predict(frame, imgsz=cfg.YOLO_IMGSZ, conf=cfg.YOLO_CONF,
                              iou=cfg.YOLO_IOU, verbose=False):
        if r.keypoints is not None and len(r.keypoints) > 0:
            return r.keypoints.data[0].cpu().numpy()
    return None

def _run_yolo_batch(self, frames):
    """
    Multi-frame YOLO inference. Used when batching is profitable
    (PyTorch backend). Falls back to per-frame loop for ONNX/OpenVINO.
    """
    if not frames:
        return []
    if not self.yolo_batch_infer or len(frames) == 1:
        return [self._run_yolo_single(f) for f in frames]
    results = self.yolo.predict(frames, imgsz=cfg.YOLO_IMGSZ, conf=cfg.YOLO_CONF,
                                iou=cfg.YOLO_IOU, verbose=False)
    out = []
    for r in results:
        if r.keypoints is not None and len(r.keypoints) > 0:
            out.append(r.keypoints.data[0].cpu().numpy())
        else:
            out.append(None)
    out.extend([None] * max(0, len(frames) - len(out)))
    return out[:len(frames)]

def _run_hands(self, frame):
    """
    Run MediaPipe Hands on a frame. Returns (left_hand_kps, right_hand_kps),
    each either an (N, 3) numpy array or None.
    Note that MediaPipe reports handedness from the camera's POV, so
    we swap labels to match the patient's POV.
    """
    h, w = frame.shape[:2]
    left_kps = right_kps = None

    def conf(p):
        for a in ("presence_score", "presence", "visibility"):
            v = getattr(p, a, None)
            if v is not None:

```

```

        try:
            return float(v)
        except (TypeError, ValueError):
            pass
    return 1.0

if self._mp_legacy:
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    res = self.hands.process(rgb)
    if res.multi_hand_landmarks and res.multi_handedness:
        for lm, hd in zip(res.multi_hand_landmarks, res.multi_handedness):
            label = hd.classification[0].label
            arr = np.array([[p.x * w, p.y * h, conf(p)] for p in lm.landmark],
dtype=np.float32)

            if label == "Left":
                right_kps = arr
            else:
                left_kps = arr
    else:
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        mp_img = mp.Image(image_format=mp.ImageFormat.SRGB, data=rgb)
        res = self.hands.detect(mp_img)
        if res.hand_landmarks and res.handedness:
            for lm_list, hd_list in zip(res.hand_landmarks, res.handedness):
                label = hd_list[0].category_name
                arr = np.array([[p.x * w, p.y * h, conf(p)] for p in lm_list],
dtype=np.float32)

                if label == "Left":
                    right_kps = arr
                else:
                    left_kps = arr

    return left_kps, right_kps

def process_batch(self, frames_by_cam):
    """
    Main entry point. Accepts a {cam_id: frame} dict, runs pose and
    hand detection respecting per-stage frame skipping, and returns
    {cam_id: {"kps", "hand_left", "hand_right", "raw_frame"}}.
    """
    if not frames_by_cam:
        return {}
    cam_ids = list(frames_by_cam.keys())
    frames = [frames_by_cam[cid] for cid in cam_ids]
    for cid in cam_ids:
        self._ensure_smoother(cid)

    # Decide which cameras need a fresh pose inference this tick.
    pose_ids, pose_frames, raw_by_cam = [], [], {}
    for cid, frame in zip(cam_ids, frames):
        tick = self._pose_tick.get(cid, 0) + 1
        self._pose_tick[cid] = tick
        if tick % self.pose_every_n_frames == 0 or cid not in self._pose_last:
            pose_ids.append(cid)
            pose_frames.append(frame)
        else:
            raw_by_cam[cid] = self._pose_last.get(cid)
    if pose_frames:
        for cid, kps in zip(pose_ids, self._run_yolo_batch(pose_frames)):
            self._pose_last[cid] = kps
            raw_by_cam[cid] = kps

    # Smooth keypoints + run hands at its own cadence.
    out = {}
    for cid, frame in zip(cam_ids, frames):
        raw = raw_by_cam.get(cid)
        kps = self.smoothers[cid].smooth(raw) if raw is not None else None
        tick = self._hand_tick.get(cid, 0) + 1
        self._hand_tick[cid] = tick
        if tick % self.hands_every_n_frames == 0 or cid not in self._hand_last:
            lk, rk = self._run_hands(frame)
            self._hand_last[cid] = (lk, rk)
        else:
            lk, rk = self._hand_last.get(cid, (None, None))
        out[cid] = {"kps": kps, "hand_left": lk, "hand_right": rk, "raw_frame": frame}
    return out

```

## ДОДАТОК В

### REST API ТА ПОТІКОВА ПЕРЕДАЧА ВІДЕОДАНИХ

```

"""
server.py - FastAPI HTTP API for the markerless goniometry subsystem.

Exposes:
* MJPEG live streams of processed frames (per camera and 3-camera grid),
* WebRTC negotiation endpoint for low-latency video delivery,
* Current angles / FPS / status,
* Session start / stop / status.

The PipelineManager singleton owns the heavy state (capture threads,
inference worker, angle engine). All endpoints serialize from it.
"""

from __future__ import annotations

import asyncio
import threading
import time
from contextlib import asynccontextmanager
from typing import Dict, Iterable, List, Optional, Set

import cv2
import numpy as np
from fastapi import FastAPI, HTTPException, Query
from fastapi.middleware.cors import CORSMiddleware
from fastapi.responses import HTMLResponse, Response, StreamingResponse
from pydantic import BaseModel

import config as cfg
from pipeline import PipelineManager

# WebRTC stack is optional. If aiortc / av are missing we fall back to
# MJPEG-only mode without raising at import time.
try:
    from aiortc import RTCPeerConnection, RTCSessionDescription, VideoStreamTrack
    from av import VideoFrame
    _WEBRTC_IMPORT_ERROR = ""
except Exception as _webrtc_exc:
    RTCPeerConnection = None # type: ignore[assignment]
    RTCSessionDescription = None # type: ignore[assignment]
    VideoStreamTrack = object # type: ignore[assignment,misc]
    VideoFrame = None # type: ignore[assignment]
    _WEBRTC_IMPORT_ERROR = str(_webrtc_exc)

# =====
# GLOBAL PIPELINE SINGLETON + COMMON HELPERS
# =====

_PIPELINE_LOCK = threading.Lock()
_PIPELINE: Optional[PipelineManager] = None
_STREAM_HEADERS = {
    "Cache-Control": "no-store, no-cache, must-revalidate, max-age=0",
    "Pragma": "no-cache",
    "Expires": "0",
    "X-Accel-Buffering": "no",
}
_WEBRTC_AVAILABLE = bool(RTCPeerConnection is not None and VideoFrame is not None)
_WEBRTC_PCS: Set = set()
_WEBRTC_PCS_LOCK = threading.Lock()

def _angle_status(value: float, low: float, high: float) -> str:
    margin = float(getattr(cfg, "ANGLE_WARN_MARGIN", 10.0))
    if (low - margin) <= value <= (high + margin):

```

```

        return "ok" if low <= value <= high else "warn"
    return "bad"

def _get_pipeline() -> PipelineManager:
    """Lazily instantiate the singleton on first request."""
    global _PIPELINE
    with _PIPELINE_LOCK:
        if _PIPELINE is None:
            _PIPELINE = PipelineManager()
        return _PIPELINE

def _camera_ids() -> List[str]:
    return [str(c["id"]) for c in cfg.CAMERAS]

def _validate_cam_id(cam_id: str) -> str:
    cam_id = str(cam_id).strip().lower()
    if cam_id not in _camera_ids():
        raise HTTPException(status_code=404, detail=f"Unknown camera '{cam_id}'")
    return cam_id

def _blank_frame() -> np.ndarray:
    """Placeholder used while no frame is available."""
    frame = np.zeros((int(cfg.FRAME_H), int(cfg.FRAME_W), 3), dtype=np.uint8)
    cv2.putText(frame, "Waiting for stream...", (18, 40),
                cv2.FONT_HERSHEY_SIMPLEX, 1.0, (80, 200, 255), 2, cv2.LINE_AA)
    return frame

def _latest_result(wait_s: float = 1.2) -> dict:
    """
    Block-with-timeout until the pipeline produces at least one frame.
    Used by /api/v1/* GETs so that the very first request after server
    boot doesn't return 'no data' just because inference hasn't ticked.
    """
    pm = _get_pipeline()
    end = time.time() + max(0.0, float(wait_s))
    latest = pm.get_latest()
    while time.time() < end:
        if latest.get("annotated"):
            return latest
        time.sleep(0.03)
        latest = pm.get_latest()
    return latest

def _frame_for_cam(cam_id: str) -> np.ndarray:
    result = _latest_result()
    annotated = result.get("annotated") or {}
    frame = annotated.get(cam_id)
    return frame if frame is not None else _blank_frame()

def _frame_for_cam_nowait(cam_id: str) -> np.ndarray:
    """Streaming path: always return the newest frame immediately."""
    result = _get_pipeline().get_latest()
    annotated = result.get("annotated") or {}
    frame = annotated.get(cam_id)
    return frame if frame is not None else _blank_frame()

def _encode_jpeg(frame: np.ndarray, quality: int) -> bytes:
    ok, buf = cv2.imencode(".jpg", frame, [int(cv2.IMWRITE_JPEG_QUALITY), int(quality)])
    if not ok:
        raise HTTPException(status_code=500, detail="JPEG encoding failed")

```

```

return bytes(buf)

def _serialize_angles(angles: Iterable, include_missing: bool = False) -> List[dict]:
    """
    Convert AngleResult instances to JSON dicts. When include_missing
    is True, missing joints are filled with neutral defaults so the web
    UI keeps a stable row layout across frames.
    """
    by_name: Dict[str, dict] = {}
    for a in angles:
        lo, hi = a.normal_range
        by_name[str(a.name)] = {
            "name": str(a.name),
            "value_deg": float(a.value),
            "unit": "degrees",
            "status": str(a.status),
            "source_cam": str(a.source_cam),
            "normal_low": float(lo),
            "normal_high": float(hi),
        }

    if not include_missing:
        return list(by_name.values())

    # Pad with neutral entries for joints not currently detected.
    items: List[dict] = []
    template_names = set()
    for base_name, rng in dict(getattr(cfg, "JOINT_RANGES", {})).items():
        for side in ("L", "R"):
            nm = f"{base_name}{side}"
            template_names.add(nm)
            row = by_name.get(nm)
            if row is None:
                lo = float(rng[0]); hi = float(rng[1])
                row = {
                    "name": nm, "value_deg": 0.0, "unit": "degrees",
                    "status": _angle_status(0.0, lo, hi),
                    "source_cam": "-",
                    "normal_low": lo, "normal_high": hi,
                }
            items.append(row)

    for nm, row in by_name.items():
        if nm not in template_names:
            items.append(row)
    return items

def _compose_grid_frame_nowait() -> np.ndarray:
    """Stitch all camera frames horizontally for the /grid stream."""
    result = _get_pipeline().get_latest()
    annotated = result.get("annotated") or {}
    parts: List[np.ndarray] = []
    for cam in cfg.CAMERAS:
        frame = annotated.get(cam["id"])
        if frame is None:
            frame = _blank_frame()
        parts.append(cv2.resize(frame, (int(cfg.FRAME_W), int(cfg.FRAME_H))))
    return np.hstack(parts) if parts else _blank_frame()

# =====
# WEBRTC TRACK
# =====

class WebRTCoffer(BaseModel):
    sdp: str
    type: str
    cam_id: str

```

```

if _WEBRTC_AVAILABLE:

    class CameraStreamTrack(VideoStreamTrack):
        """
        Pulls the latest processed frame for the given camera and
        publishes it as a WebRTC video track. Per-track FPS limit
        prevents heavy encode pressure when the browser is faster
        than the inference cycle.
        """

        def __init__(self, cam_id: str, max_fps: float):
            super().__init__()
            self.cam_id = str(cam_id)
            self.interval = 1.0 / max(1.0, float(max_fps))
            self._last_emit = 0.0

        async def recv(self):
            pts, time_base = await self.next_timestamp()
            now = time.perf_counter()
            if self._last_emit > 0.0:
                spare = self.interval - (now - self._last_emit)
                if spare > 0:
                    await asyncio.sleep(spare)
            self._last_emit = time.perf_counter()

            frame = _frame_for_cam_nowait(self.cam_id)
            vframe = VideoFrame.from_ndarray(frame, format="bgr24")
            vframe.pts = pts
            vframe.time_base = time_base
            return vframe

    async def _close_all_webrtc_peers() -> None:
        with _WEBRTC_PCS_LOCK:
            peers = list(_WEBRTC_PCS)
            _WEBRTC_PCS.clear()
        for pc in peers:
            try:
                await pc.close()
            except Exception:
                pass

    async def _wait_for_ice_complete(pc, timeout_s: float = 2.0) -> None:
        """Wait for trickle-ICE gathering to finish, with a hard timeout."""
        deadline = time.perf_counter() + max(0.2, float(timeout_s))
        while pc.iceGatheringState != "complete" and time.perf_counter() < deadline:
            await asyncio.sleep(0.05)

    # =====
    # APP LIFESPAN
    # =====

    @asynccontextmanager
    async def _lifespan(_: FastAPI):
        """Start the pipeline on app startup, stop it on shutdown."""
        pm = _get_pipeline()
        pm.start()
        try:
            yield
        finally:
            if _WEBRTC_AVAILABLE:
                await _close_all_webrtc_peers()
            pm.stop()

    app = FastAPI(title="Pose Tracker API", version="1.0.0", lifespan=_lifespan)

```

```

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"], allow_credentials=True,
    allow_methods=["*"], allow_headers=["*"],
)

# =====
# STATUS / METADATA ENDPOINTS
# =====

@app.get("/api/v1/health")
def health():
    pm = _get_pipeline()
    latest = pm.get_latest()
    return {
        "status": "ok",
        "has_frames": bool((latest.get("annotated") or {})),
        "timestamp_unix": time.time(),
    }

@app.get("/api/v1/cameras")
def cameras():
    return {
        "items": [
            {
                "id": str(c["id"]),
                "name": str(c.get("name", c["id"])),
                "view": str(c.get("view", "")),
            }
            for c in cfg.CAMERAS
        ]
    }

@app.get("/api/v1/status")
def status():
    pm = _get_pipeline()
    latest = pm.get_latest()
    live = latest.get("live_flags") or {}
    fps_map = latest.get("fps_map") or {}
    session = pm.get_session_status()
    total = len(cfg.CAMERAS)
    online = sum(1 for v in live.values() if v)
    return {
        "online_cameras": online,
        "total_cameras": total,
        "avg_fps": float(latest.get("avg_fps") or 0.0),
        "fps_map": {k: float(v) for k, v in fps_map.items()},
        "live_flags": {k: bool(v) for k, v in live.items()},
        "session": session,
        "timestamp_unix": time.time(),
    }

@app.get("/api/v1/angles")
def angles():
    latest = _latest_result()
    angles_arr = _serialize_angles(latest.get("angles") or [], include_missing=True)
    return {
        "count": len(angles_arr),
        "avg_fps": float(latest.get("avg_fps") or 0.0),
        "items": angles_arr,
        "timestamp_unix": time.time(),
    }

# =====
# STILL-FRAME ENDPOINTS
# =====

```

```

@app.get("/api/v1/frame/{cam_id}.jpg")
def frame_jpg(cam_id: str, quality: int = Query(default=82, ge=30, le=100)):
    cam_id = _validate_cam_id(cam_id)
    frame = _frame_for_cam(cam_id)
    jpg = _encode_jpeg(frame, quality=quality)
    return Response(content=jpg, media_type="image/jpeg",
                    headers={"Cache-Control": "no-store"})

@app.get("/api/v1/frame/grid.jpg")
def frame_grid_jpg(quality: int = Query(default=82, ge=30, le=100)):
    frame = _compose_grid_frame_nowait()
    jpg = _encode_jpeg(frame, quality=quality)
    return Response(content=jpg, media_type="image/jpeg",
                    headers={"Cache-Control": "no-store"})

# =====
# MJPEG STREAMING
# =====

def _mjpeg_stream_single(cam_id: str, quality: int, max_fps: float):
    """
    Generator that yields multipart/x-mixed-replace JPEG parts.
    Caps emission at max_fps to control bandwidth and CPU usage.
    """
    interval = 1.0 / max(1.0, max_fps)
    while True:
        tick = time.perf_counter()
        frame = _frame_for_cam_nowait(cam_id)
        jpg = _encode_jpeg(frame, quality=quality)
        yield (
            b"--frame\r\n"
            b"Content-Type: image/jpeg\r\n"
            b"Cache-Control: no-store\r\n"
            b"Content-Length: " + str(len(jpg)).encode("ascii") + b"\r\n\r\n" + jpg + b"\r\n"
        )
        spare = interval - (time.perf_counter() - tick)
        if spare > 0:
            time.sleep(spare)

def _mjpeg_stream_grid(quality: int, max_fps: float):
    """Same as the per-camera stream but emits the stitched 3-up grid."""
    interval = 1.0 / max(1.0, max_fps)
    while True:
        tick = time.perf_counter()
        frame = _compose_grid_frame_nowait()
        jpg = _encode_jpeg(frame, quality=quality)
        yield (
            b"--frame\r\n"
            b"Content-Type: image/jpeg\r\n"
            b"Cache-Control: no-store\r\n"
            b"Content-Length: " + str(len(jpg)).encode("ascii") + b"\r\n\r\n" + jpg + b"\r\n"
        )
        spare = interval - (time.perf_counter() - tick)
        if spare > 0:
            time.sleep(spare)

@app.get("/api/v1/stream/{cam_id}")
def stream_cam(
    cam_id: str,
    quality: int = Query(default=int(getattr(cfg, "API_STREAM_QUALITY", 68)), ge=30, le=100),
    max_fps: float = Query(default=float(getattr(cfg, "API_STREAM_MAX_FPS", 20.0)), ge=1.0,
le=30.0),
):
    cam_id = _validate_cam_id(cam_id)
    return StreamingResponse(
        _mjpeg_stream_single(cam_id=cam_id, quality=quality, max_fps=max_fps),
        media_type="multipart/x-mixed-replace; boundary=frame",

```

```

        headers=_STREAM_HEADERS,
    )

@app.get("/api/v1/stream/grid")
def stream_grid(
    quality: int = Query(default=int(getattr(cfg, "API_STREAM_QUALITY", 68)), ge=30, le=100),
    max_fps: float = Query(default=float(getattr(cfg, "API_STREAM_GRID_MAX_FPS", 15.0)), ge=1.0,
le=30.0),
):
    return StreamingResponse(
        _mjpeg_stream_grid(quality=quality, max_fps=max_fps),
        media_type="multipart/x-mixed-replace; boundary=frame",
        headers=_STREAM_HEADERS,
    )

# =====
# WEBRTC OFFER/ANSWER ENDPOINT
# =====

@app.post("/api/v1/webrtc/offer")
async def webrtc_offer(payload: WebRTCoffer):
    """
    Standard WebRTC SDP offer/answer exchange.
    The browser sends its offer, the server attaches a CameraStreamTrack
    for the requested camera, and returns its SDP answer. ICE gathering
    is awaited synchronously to keep the API simple (no trickle).
    """
    if not _WEBRTC_AVAILABLE:
        detail = "WebRTC backend is unavailable. Install aiortc and av."
        if _WEBRTC_IMPORT_ERROR:
            detail += f" Import error: {_WEBRTC_IMPORT_ERROR}"
        raise HTTPException(status_code=503, detail=detail)

    cam_id = _validate_cam_id(payload.cam_id)
    if str(payload.type).lower() != "offer":
        raise HTTPException(status_code=400, detail="SDP type must be 'offer'")

    pc = RTCPeerConnection()
    with _WEBRTC_PCS_LOCK:
        _WEBRTC_PCS.add(pc)

    @pc.on("connectionstatechange")
    async def on_connectionstatechange():
        # Self-cleanup when the browser disconnects.
        if pc.connectionState in {"failed", "closed", "disconnected"}:
            with _WEBRTC_PCS_LOCK:
                _WEBRTC_PCS.discard(pc)
            await pc.close()

    track = CameraStreamTrack(
        cam_id=cam_id,
        max_fps=float(getattr(cfg, "WEBRTC_MAX_FPS",
getattr(cfg, "API_STREAM_MAX_FPS", 20.0))),
    )
    pc.addTrack(track)

    try:
        await pc.setRemoteDescription(RTCSessionDescription(sdp=str(payload.sdp),
type="offer"))
        answer = await pc.createAnswer()
        await pc.setLocalDescription(answer)
        await _wait_for_ice_complete(pc, timeout_s=2.0)
        local = pc.localDescription
        if local is None:
            raise RuntimeError("Failed to create WebRTC local description")
        return {"type": local.type, "sdp": local.sdp}
    except HTTPException:
        raise
    except Exception as exc:
        with _WEBRTC_PCS_LOCK:

```

```

        _WEBRTC_PCS.discard(pc)
    try:
        await pc.close()
    except Exception:
        pass
    raise HTTPException(status_code=500, detail=f"WebRTC negotiation failed: {exc}")

# =====
#  SESSION CONTROL ENDPOINTS
# =====

@app.post("/api/v1/session/start")
def session_start():
    pm = _get_pipeline()
    sid = pm.start_session()
    return {"ok": True, "session_id": sid, "status": pm.get_session_status()}

@app.post("/api/v1/session/stop")
def session_stop():
    pm = _get_pipeline()
    result = pm.stop_session()
    if result.get("error"):
        raise HTTPException(status_code=409, detail=result["error"])
    return {"ok": True, "result": result, "status": pm.get_session_status()}

@app.get("/api/v1/session/status")
def session_status():
    pm = _get_pipeline()
    return pm.get_session_status()

# =====
#  RUNTIME CONFIG REPORT
# =====

@app.get("/api/v1/config")
def api_config():
    return {
        "frame_w": int(cfg.FRAME_W),
        "frame_h": int(cfg.FRAME_H),
        "target_fps": int(cfg.TARGET_FPS),
        "rtsp_buffer": int(cfg.RTSP_BUFFER),
        "yolo_runtime": str(getattr(cfg, "YOLO_RUNTIME", "torch")),
        "yolo_imgsz": int(getattr(cfg, "YOLO_IMGSZ", 320)),
        "hands_every_n_frames": int(getattr(cfg, "HANDS_EVERY_N_FRAMES", 1)),
        "pose_every_n_frames": int(getattr(cfg, "POSE_EVERY_N_FRAMES", 1)),
        "webrtc_enabled": bool(_WEBRTC_AVAILABLE),
        "cameras": [
            {"id": str(c["id"]),
             "name": str(c.get("name", c["id"])),
             "view": str(c.get("view", ""))}
            for c in cfg.CAMERAS
        ],
    }

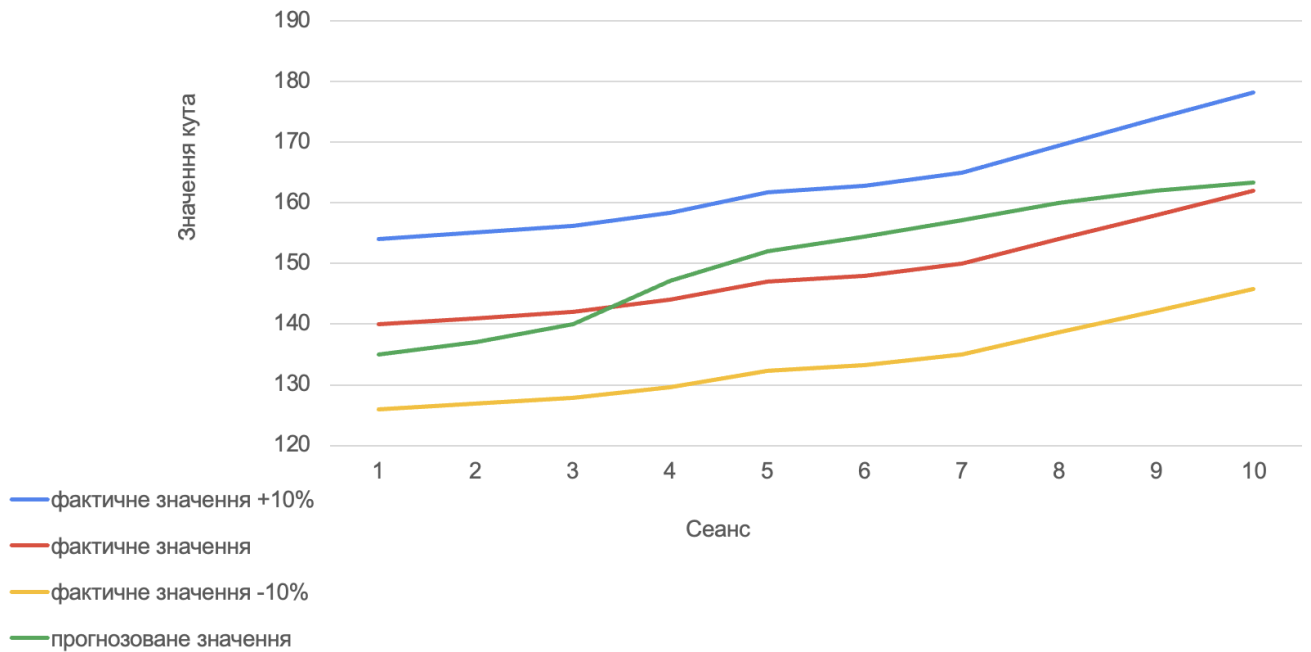
def run_api(host: str = "0.0.0.0", port: int = 8080):
    import uvicorn
    uvicorn.run("server:app", host=host, port=port, reload=False, log_level="info")

if __name__ == "__main__":
    run_api()

```

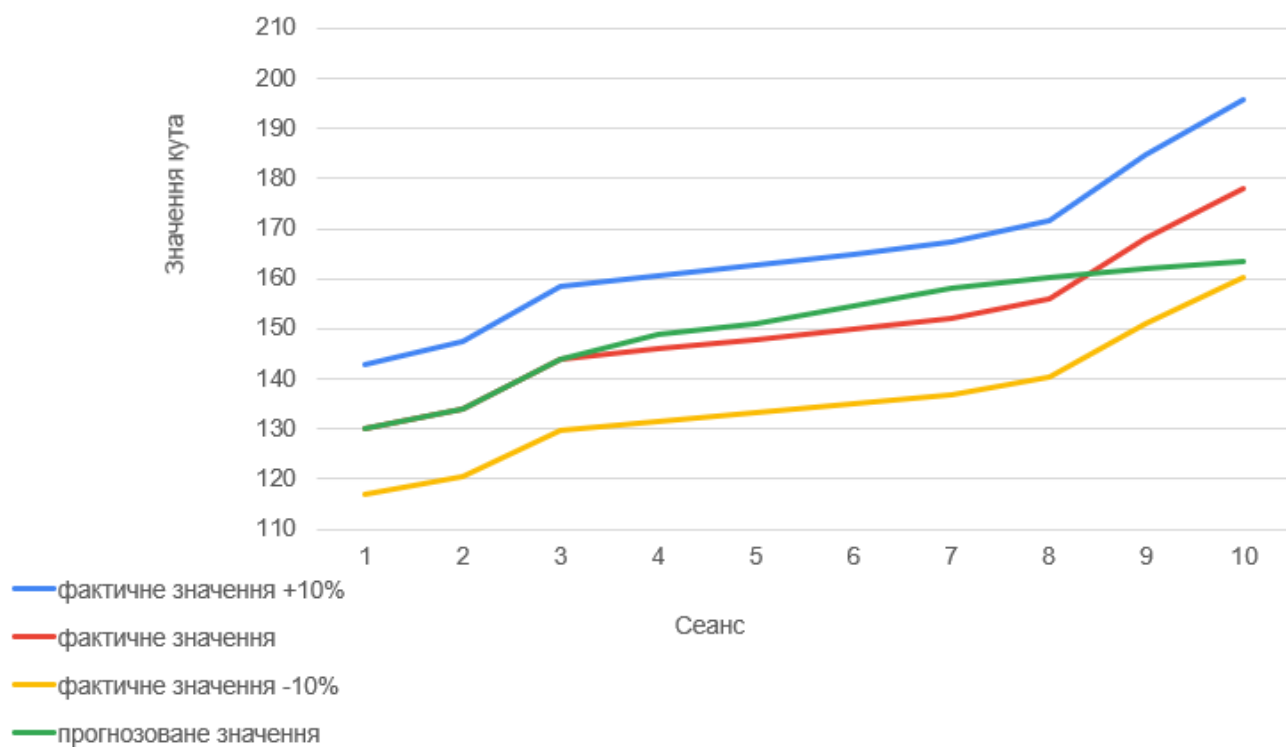
**ДОДАТОК Г**  
**РЕЗУЛЬТАТИ ІДЕНТИФІКАЦІЇ ІНТЕРВАЛЬНОЇ МОДЕЛІ ДИНАМІКИ**  
**ВІДНОВЛЕННЯ КУТА ЗГИНАННЯ ПЛЕЧОВОГО СУГЛОБА**

Анастасія



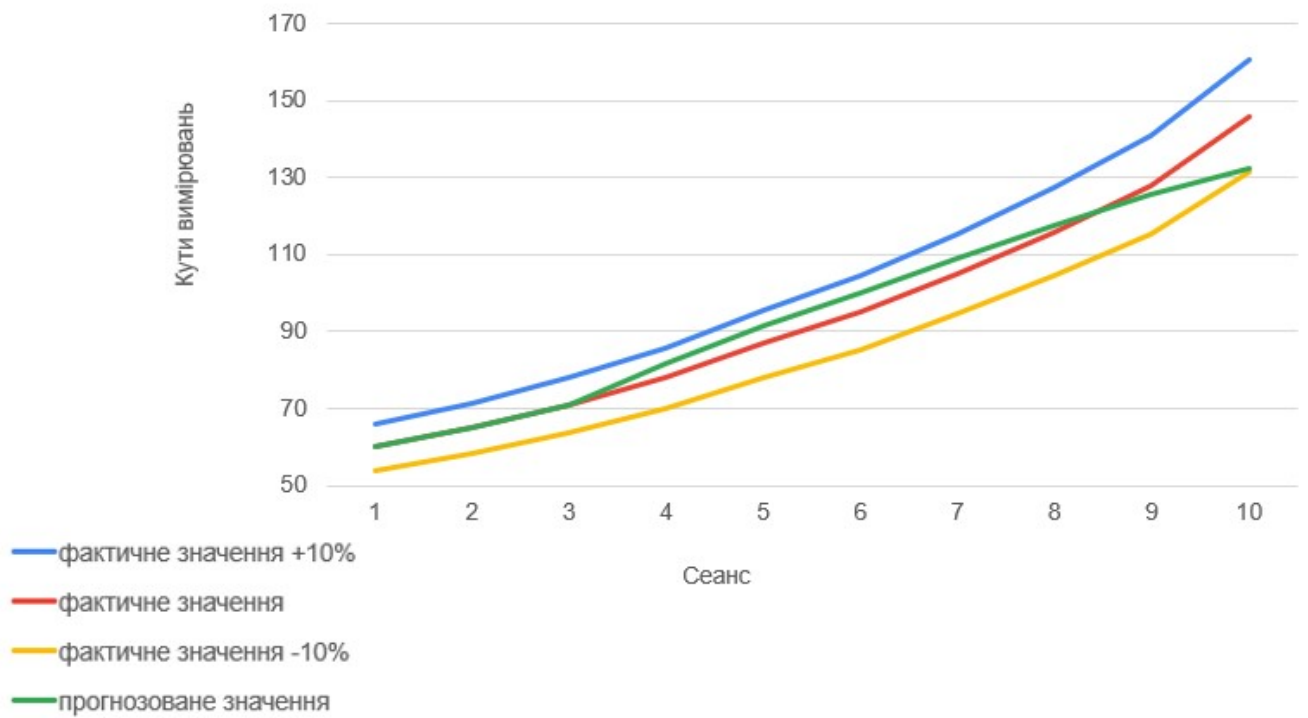
сеанс	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	154	155	156	158	162	163	165	169	174	178
фактичне значення	140	141	142	144	147	148	150	154	158	162
фактичне значення -10%	126	127	128	130	132	133	135	139	142	146
прогнозоване значення	135	137	140	147	152	154	157	160	162	163

## Валентина



	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	143	147	158	161	163	165	167	172	185	196
фактичне значення	130	134	144	146	148	150	152	156	168	178
фактичне значення -10%	117	121	130	131	133	135	137	140	151	160
прогнозоване значення	130	134	144	149	151	155	158	160	162	164

## Сергій

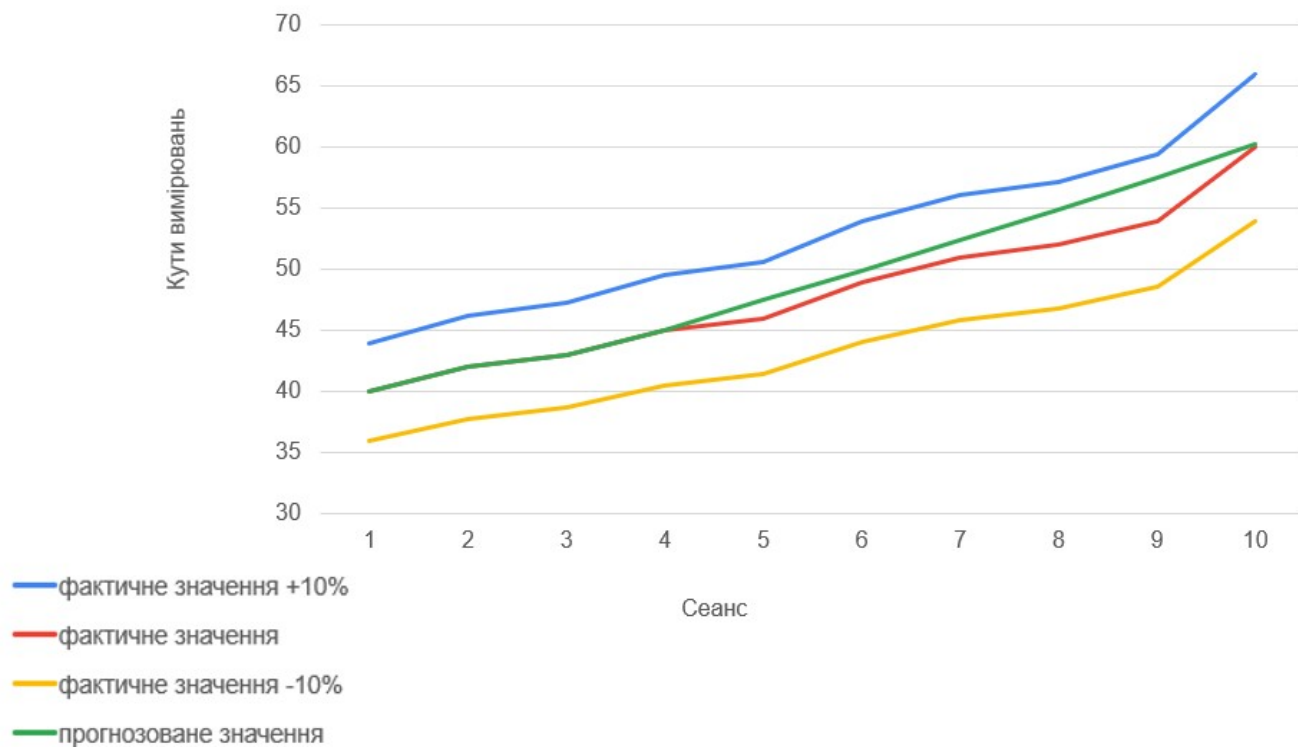


	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	66	72	78	86	96	105	116	128	141	161
фактичне значення	60	65	71	78	87	95	105	116	128	146
фактичне значення -10%	54	59	64	70	78	86	95	104	115	131
прогнозоване значення	60	65	71	82	92	100	109	118	125	132

## ДОДАТОК Г

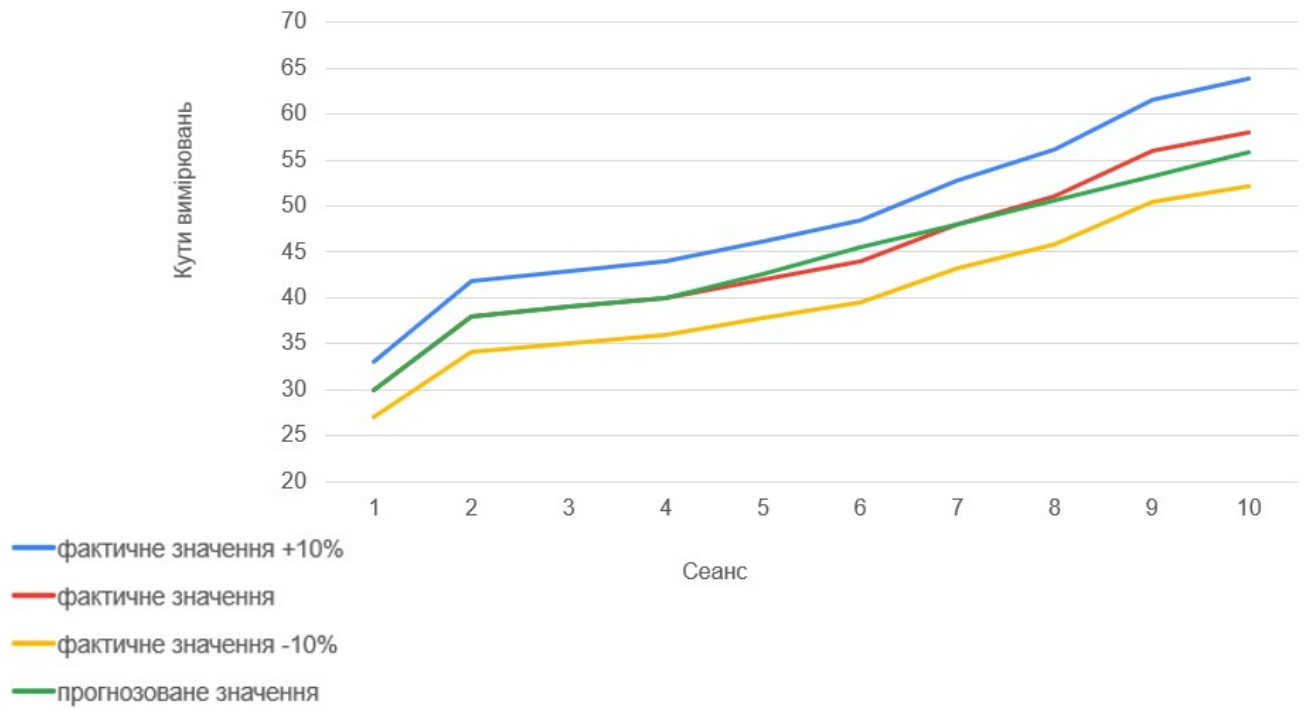
### РЕЗУЛЬТАТИ ІДЕНТИФІКАЦІЇ ІНТЕРВАЛЬНОЇ МОДЕЛІ ДИНАМІКИ ВІДНОВЛЕННЯ КУТА РОЗГИНАННЯ ПЛЕЧОВОГО СУГЛОБА

Анастасія



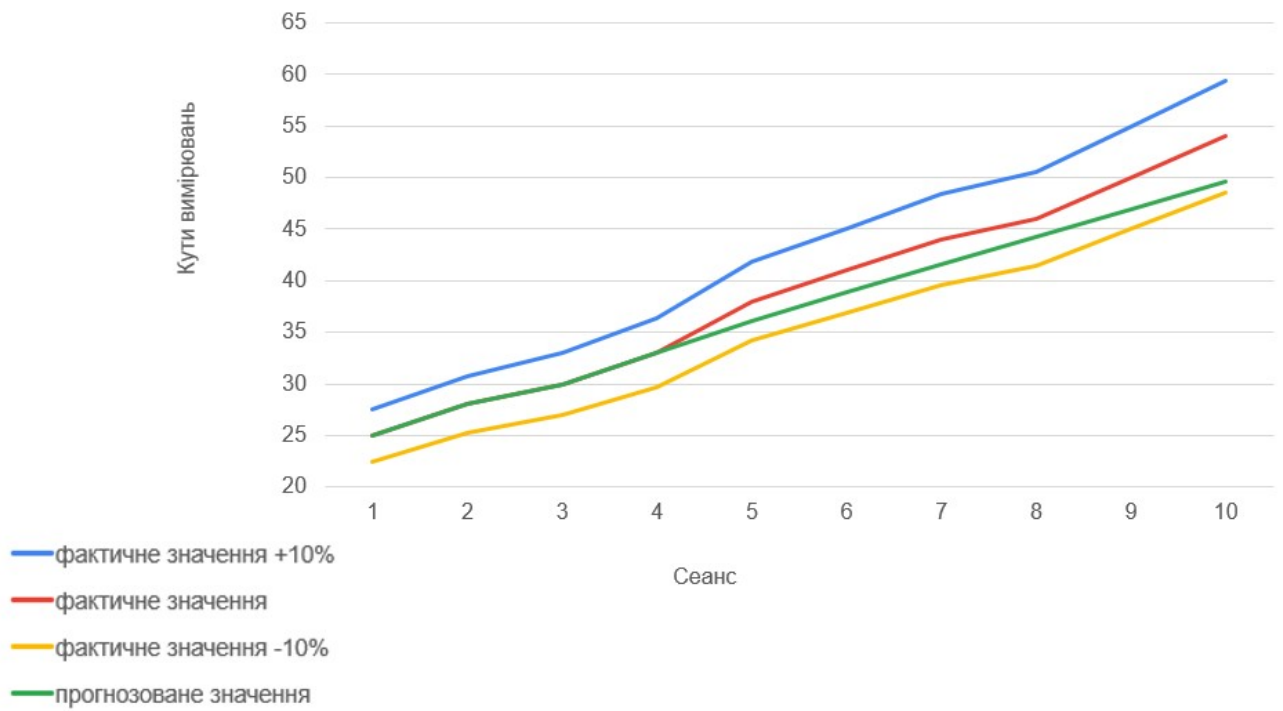
	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	44	46	47	50	51	54	56	57	59	66
фактичне значення	40	42	43	45	46	49	51	52	54	60
фактичне значення -10%	36	38	39	41	41	44	46	47	49	54
прогнозоване значення	40	42	43	45	47	50	52	55	58	60

## Валентина



	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	33	42	43	44	46	48	53	56	62	64
фактичне значення	30	38	39	40	42	44	48	51	56	58
фактичне значення -10%	27	34	35	36	38	40	43	46	50	52
прогнозоване значення	30	38	39	40	43	45	48	51	53	56

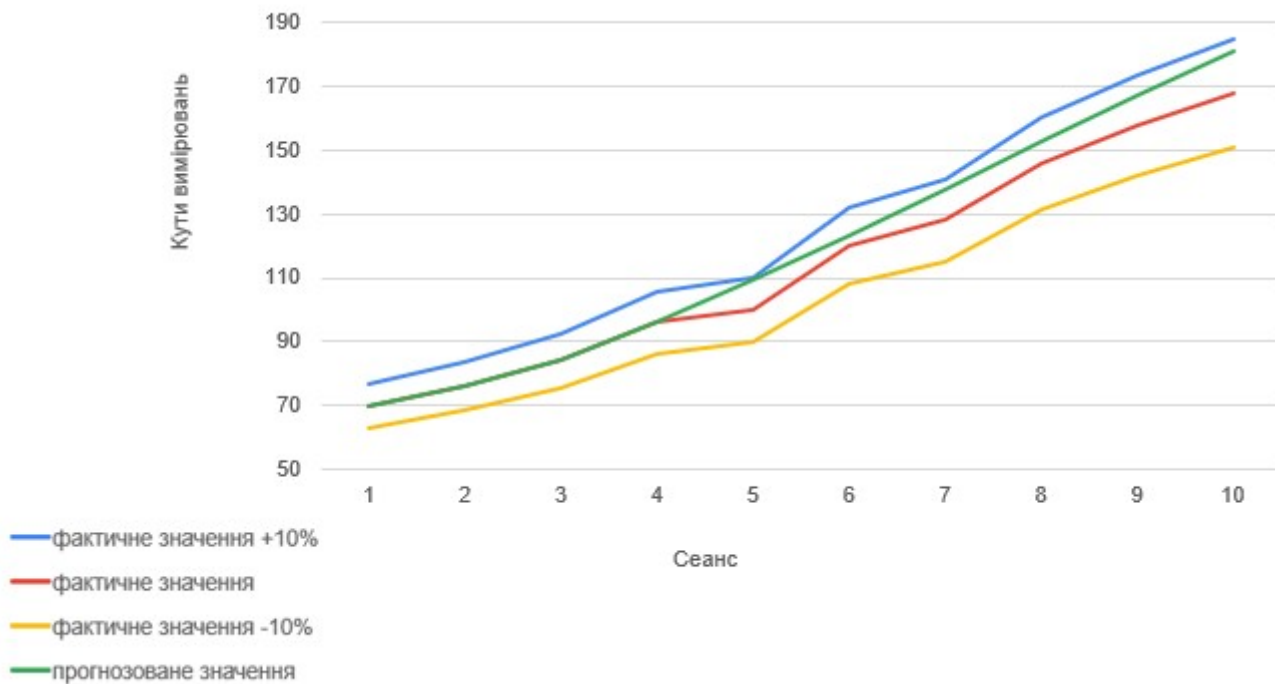
## Сергій



	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	28	31	33	36	42	45	48	51	55	59
фактичне значення	25	28	30	33	38	41	44	46	50	54
фактичне значення -10%	23	25	27	30	34	37	40	41	45	49
прогнозоване значення	25	28	30	33	36	39	42	44	47	50

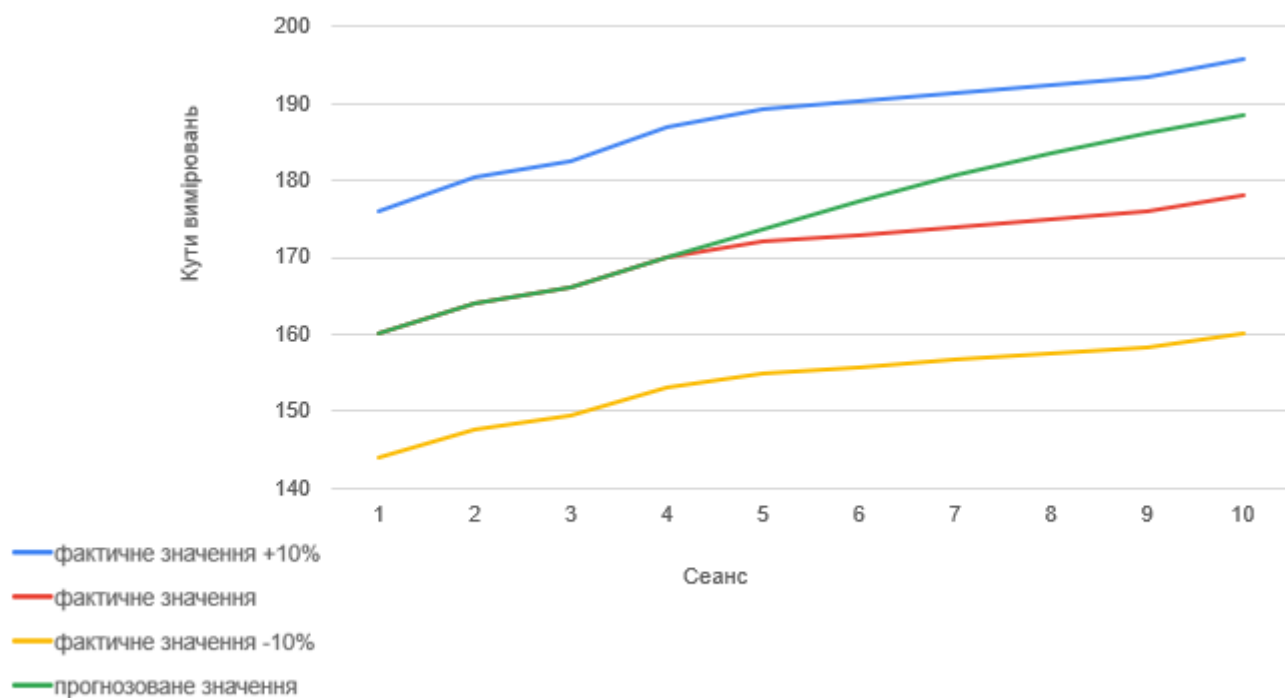
**ДОДАТОК Д**  
**РЕЗУЛЬТАТИ ІДЕНТИФІКАЦІЇ ІНТЕРВАЛЬНОЇ МОДЕЛІ ДИНАМІКИ**  
**ВІДНОВЛЕННЯ КУТА ВІДВЕДЕННЯ ПЛЕЧОВОГО СУГЛОБА**

Анастасія



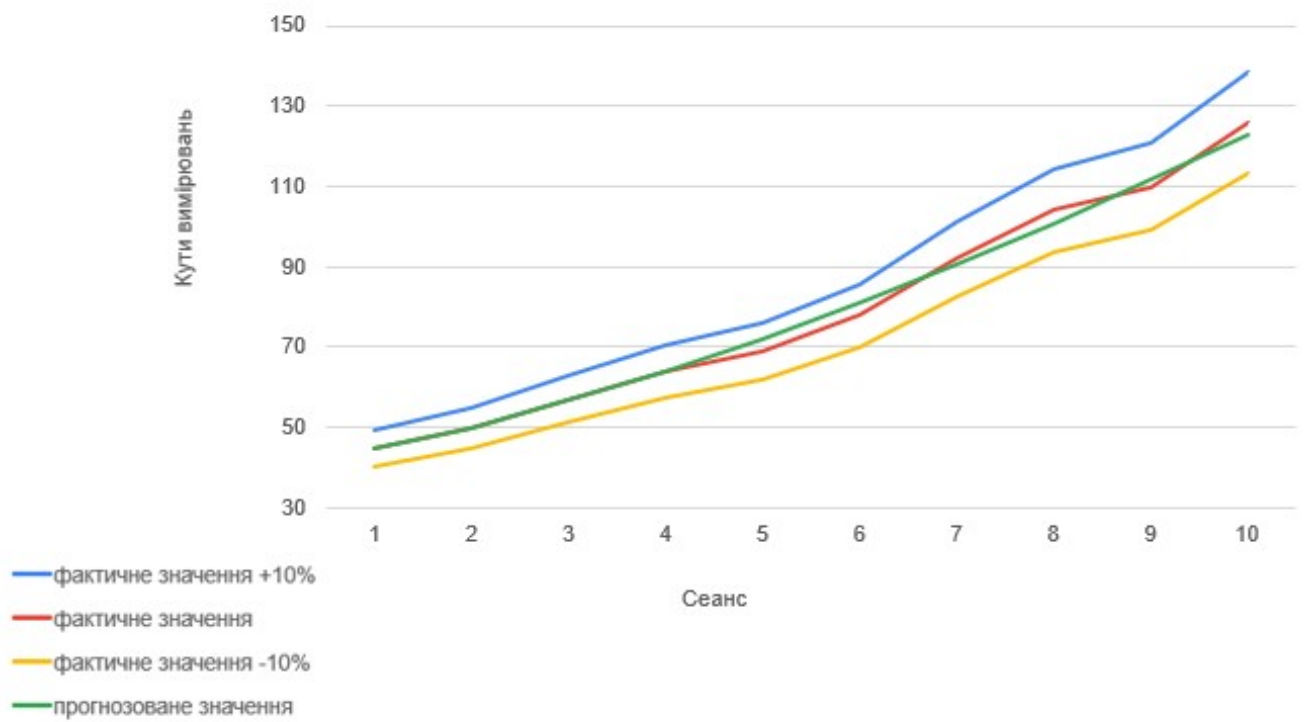
	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	77	84	92	106	110	132	141	161	174	185
фактичне значення	70	76	84	96	100	120	128	146	158	168
фактичне значення -10%	63	68	76	86	90	108	115	131	142	151
прогнозоване значення	70	76	84	96	109	123	138	153	167	181

## Валентина



	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	176	180	183	187	189	190	191	193	194	196
фактичне значення	160	164	166	170	172	173	174	175	176	178
фактичне значення -10%	144	148	149	153	155	156	157	158	158	160
прогнозоване значення	160	164	166	170	174	177	181	184	186	188

## Сергій



	1	2	3	4	5	6	7	8	9	10
фактичне значення +10%	50	55	63	70	76	86	101	114	121	139
фактичне значення	45	50	57	64	69	78	92	104	110	126
фактичне значення -10%	41	45	51	58	62	70	83	94	99	113
прогнозоване значення	45	50	57	64	72	81	91	101	112	123

## ДОДАТОК Е

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. **Tsapiv Y.**, Dyvak M., Pukas A., Petrovskyi Y., Melnyk A., Dyvak A., Banasik A., Czupryna-Nowak A., Pikiewicz P., Popyk Y., Dzyha Y. AR Technology for Restoring Upper Limb Joint Mobility in Patients. *Applied Sciences*, 2026, Vol. 17 (подано до друку). ISSN 2076-3417.

(Scopus, Q2).

2. Пукас А. В., **Цапів Я. А.** Автоматизована безмаркерна система вимірювання діапазонів рухів суглобів на основі трикамерного відеоаналізу. Наукові праці ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка», Дрогобич, 2026, No. 1 (42), с. 54–64. ISSN 1996-1588.

DOI: 10.31474/1996-1588-2026-1-42-54-64.

3. **Цапів Я. А.**, Пукас А. В., Біловус Д. Рекомендаційна система підтримки фізичної реабілітації на основі корекції терапевтичних зображень. Вимірювальна та обчислювальна техніка в технологічних процесах, Хмельницький, 2026 (подано до друку). ISSN 2219-9365.

4. **Цапів Я. А.** Адаптивний цикл реабілітації верхніх кінцівок на основі AR-технологій. Комп'ютерні інформаційні технології: Матеріали весняної школи-семінару молодих вчених і студентів СІТ'2026. – Тернопіль: ЗУНУ, 2026, с. 78–79.

5. **Tsapiv Y.**, Dyvak M., Tykhyi R., Popovych D., Havrylenko A., Kasianchuk V. Automated goniometric analysis system based on neural networks. *Proceedings - International Conference on Advanced Computer Information Technologies, ACIT*, 2025, Sibenik, Croatia, p. 762–766. ISSN 2770-5218.

DOI: <https://doi.org/10.1109/ACIT65614.2025.11185805> (Scopus).

6. **Цапів Я. А.**, Тихий Р. Р. Інноваційна вебсистема гоніометричного аналізу на базі нейронних мереж. Наукові праці ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка», Дрогобич, 2025, с. 80–88. ISSN 1996-1588.

DOI: 10.31474/1996-1588-2025-1-40-80-88.

7. **Цапів Я. А.** Архітектура програмної системи поєднання арт-терапії та фізичної реабілітації з використанням технології доповненої реальності. Комп'ютерні інформаційні технології: Матеріали зимової школи-семінару молодих вчених і студентів СІТ'2025. – Тернопіль: ЗУНУ, 2025, с. 57–58.

8. Tykhyi R., **Tsapiv Y.**, Dyvak M., Popovych D., Havrylenko A., Husieva T. Software system for supporting art therapy processes using augmented reality technologies. Proceedings - International Conference on Advanced Computer Information Technologies, ACIT, 2024, p. 660–664. ISSN 2770-5218.

DOI: <https://doi.org/10.1109/ACIT62333.2024.10712584> (Scopus).

9. **Цапів Я. А.**, Головка Р. А., Шостопаль В. А. Математичне та програмне забезпечення для організації процесу тестування web-орієнтованих систем. Комп'ютерні інформаційні технології: Матеріали школи-семінару молодих вчених і студентів СІТ'2023. – Тернопіль: ЗУНУ, 2023, с. 90.

# ДОДАТОК Є

## АКТИ ВПРОВАДЖЕНЬ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ

«ЗАТВЕРДЖУЮ»

Генеральний директор КНП  
«Тернопільська обласна  
клінічна лікарня» ТОВ  
Заслужений лікар України,  
доктор медичних наук



Бліхар В.Є.

« 11 » травня 2026 р.

### АКТ

про впровадження результатів дисертаційної роботи  
Цапів Ярослава Андрійовича

**«Математичне та програмне забезпечення реабілітаційної арт-терапії верхніх  
кінцівок пацієнтів на основі AR-технологій»**

У процесі проведення реабілітаційних сеансів для пацієнтів із порушеннями рухливості верхніх кінцівок використано результати дисертаційної роботи аспіранта кафедри комп'ютерних наук Західноукраїнського національного університету Цапів Ярослава Андрійовича, а саме:

– метод безмаркерної гоніометрії на основі аналізу трьох відеопотоків, який базується на нейромережевому детекторі YOLOv8-pose, адаптивній фільтрації координат OpenEuro-фільтром та алгоритмі геометричного злиття вимірювань від трьох ортогонально розташованих камер із пріоритизацією на основі площин руху, що забезпечує автоматизоване об'єктивне вимірювання кутів рухливості суглобів з допустимою похибкою менше 4°;

– комплексне програмне забезпечення системи реабілітаційної арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій, що реалізує замкнений адаптивний цикл керування реабілітаційним процесом на основі інтеграції автоматизованої безмаркерної гоніометрії, інтервального прогнозування динаміки реабілітації, арт-терапії на AR-технології та рекомендаційної системи з генерацією персоналізованого терапевтичного контенту.

Під час використання програмного забезпечення системи AR-реабілітації у процесі проведення реабілітаційних сеансів забезпечено скорочення курсу реабілітації в середньому на 12% завдяки адаптивній корекції терапевтичного протоколу на основі інтервального прогнозування динаміки відновлення.

Відповідальний за впровадження,  
завідуюча реабілітаційним відділенням

Брикса Н. Я.

**ЗАТВЕРДЖУЮ**Проректор закладу вищої освіти  
з наукової роботиТернопільського національного  
медичного університету  
ім. І.Я. Горбачевського МОЗ України

Олександра ОЛЕЩУК

«15» травня 2026 р.

**АКТ**

про впровадження результатів дисертаційного дослідження

Цапів Ярослава Андрійовича

**«Математичне та програмне забезпечення реабілітаційної арт-терапії  
верхніх кінцівок пацієнтів на основі AR-технологій»**

Даним актом засвідчується, що при виконанні науково-дослідної роботи за спільним проєктом кафедри фізичної терапії, ерготерапії та фізичного виховання Тернопільського національного медичного університету імені І.Я. Горбачевського Міністерства охорони здоров'я України та кафедри комп'ютерних наук Західноукраїнського національного університету впроваджено такі результати дисертаційного дослідження аспіранта ЗУНУ Цапів Ярослава Андрійовича:

– спосіб визначення кутів суглобової рухливості без застосування маркерів, що ґрунтується на паралельній обробці відеоданих із трьох камер за допомогою нейромережевої архітектури YOLOv8-pose із подальшим згладжуванням траєкторій ключових точок OpenCv-фільтром і алгоритмом фузії результатів вимірювань від трьох взаємно перпендикулярних ракурсів з урахуванням пріоритетності площин руху, що дає змогу автоматично визначати суглобові кути з точністю клінічно допустимою точністю;

– програмний комплекс для здійснення реабілітації верхніх кінцівок засобами арт-терапії в середовищі доповненої реальності, в основу якого покладено замкнений цикл адаптивного керування процесом реабілітації, що поєднує модуль автоматичної безмаркерної гоніометрії, блок прогнозування перебігу реабілітації на основі інтервальних моделей, AR-компонент арт-терапевтичних вправ та підсистему формування персоналізованих терапевтичних завдань із використанням генеративного штучного інтелекту.

Відповідальний за впровадження,

Завідувачка кафедри фізичної терапії, ерготерапії  
та фізичного вихованняТернопільського національного медичного університету  
імені І.Я. Горбачевського МОЗ України

д.мед.н, професор

Дарія ПОПОВИЧ

ТОВ «ЕЛЕКС»  
вул. Оболоня 49,  
м. Тернопіль 46008 Україна  
тел.: +380-94-840-0096,  
+380-35-240-0096  
«11» травня 2026 р.

### ДОВІДКА

про впровадження у виробничий процес Тернопільської філії  
компанії ТОВ «ЕЛЕКС» результатів дисертаційної роботи  
ЦАПІВА Ярослава Андрійовича  
«Математичне та програмне забезпечення реабілітаційної арт-терапії верхніх  
кінцівок пацієнтів на основі AR-технологій»

Даний акт складений про те, що результати дисертаційної роботи ЦАПІВА Ярослава Андрійовича на тему: «Математичне та програмне забезпечення реабілітаційної арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій» використані у виробничому процесі Тернопільської філії компанії ТОВ «ЕЛЕКС» при розробці програмних рішень у сфері комп'ютерного зору, доповненої реальності та інтелектуального аналізу даних.

При розробці модулів відеоаналізу та систем трекінгу руху використовуються метод безмаркерного трикамерного вимірювання кутів суглобів з адаптивною фільтрацією на основі фільтра OneEuro, а також підходи до інтеграції нейромережових моделей оцінки пози (YOLOv8-pose, MediaPipe Pose Landmarker) у продуктивні рішення. Метод злиття даних з декількох камер та адаптивна фільтрація шуму застосовуються в проектах, пов'язаних з обробкою відеопотоків у реальному часі.

Архітектурні рішення побудови замкнутого адаптивного циклу, що інтегрує підсистему комп'ютерного зору, модуль математичного прогнозування на основі інтервальних моделей та генеративний AI-компонент, впроваджені як референтна архітектура при проектуванні клієнтських рішень у галузі цифрової медицини та AR/VR-застосунків.

Директор Тернопільського  
відділення ТОВ «ЕЛЕКС»



Сергій КУТУЗОВ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
**НАУКОВО-ДОСЛІДНА ЧАСТИНА**

46009, Україна, м. Тернопіль, вул. Львівська, буд. 5А, тел. (0352) 51-75-52

№ 21/ 85-2026

«12» 05 2026 року

**ДОВІДКА**  
**про участь у виконанні науково-дослідних робіт**

Видана ЦАПВУ Ярославу Андрійовичу про участь у виконанні науково-дослідних робіт Західноукраїнського національного університету. Зокрема, у 2024 році на посаді молодшого наукового співробітника він долучився до виконання молодіжного держбюджетного дослідження «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (державний реєстраційний номер 0124U000076); у 2026 році на посаді наукового співробітника – до виконання держбюджетного дослідження «Інформаційна технологія забезпечення ефективної акумуляції електроенергії гібридною електростанцією» (0126U002209) та молодіжного держбюджетного дослідження «Математичне та програмне забезпечення функціонування вискоелективної гібридної когенераційної установки» (0126U002213).

Начальник  
науково-дослідної частини



Віта СЕМАНЮК



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

вул. Львівська, 11, м. Тернопіль, 46009; тел./факс +380 (352) 51-75-75;  
www.wunu.edu.ua; rektor@wunu.edu.ua; ідентифікаційний код за ЄДРПОУ 33680120



ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи  
Західноукраїнського національного  
університету, к.е.н., доцент

Віктор ОСТРОВЕРХОВ

про впровадження в освітній процес Західноукраїнського національного  
університету результатів дисертаційної роботи

**Цапів Ярослава Андрійовича**

на тему: «Математичне та програмне забезпечення реабілітаційної  
арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій»

Даний акт складений про те, що результати дисертаційної роботи здобувача ступеня доктора філософії Цапів Ярослава Андрійовича на тему: «Математичне та програмне забезпечення реабілітаційної арт-терапії верхніх кінцівок пацієнтів на основі AR-технологій» використані в освітньому процесі кафедри комп'ютерних наук факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету, зокрема під час підготовки студентів спеціальності F2 (121) – Інженерія програмного забезпечення. При викладанні дисципліни «Архітектура та проектування програмного забезпечення» розглядаються методи побудови архітектури комплексних програмних систем реального часу та організації багатокomпонентних систем, тоді як у рамках курсу «Інтелектуальний аналіз даних» розглядаються методи безмаркерної гоніометрії на основі нейромережових детекторів пози людини та адаптивної фільтрації сигналів, а у рамках курсу «Інтервальні обчислення» розглядаються інтервальні дискретні динамічні моделі відновлення кутів рухливості верхніх кінцівок пацієнтів.

Декан факультету комп'ютерних  
інформаційних технологій,  
д.т.н., доцент

Ігор ЯКИМЕНКО

Доцент кафедри комп'ютерних наук,  
к.т.н., доцент

ЗУНУ

Ірина ВОЙТЮК

№ 126-27/1399 від 03.06.2026



Витяг з протоколу № 85  
засідання комісії з біоетики Тернопільського національного медичного  
університету імені І.Я. Горбачевського МОЗ України  
від 1 квітня 2026 року

**Присутні:** проф. С.Н. Вадзюк (голова), проф. В.Б. Гощинський, проф. П.Р. Сельський, проф. І.Я. Дзюбановський, директор обласного еколого-натуралістичного центру І.І. Герц, О.М. Ратинська (секретар комісії).

**Розглянули:** матеріали використання гоніометричного аналізу та використання AR- системи для реабілітації пацієнтів із порушенням рухливості верхніх кінцівок. Автори: Попович Д.В., Дивак М.П., Цапів Я.А., Мельник А.М., Гавриленко А.В., Гусева Т.П., Ковбасюк І.С.

**Постановили:** в матеріалах використання гоніометричного аналізу та використання AR- системи для реабілітації пацієнтів із порушенням рухливості верхніх кінцівок дотримані норми та принципи біоетики, відображені в: основних положення GLP (1981р.), правилах проведення робіт з використанням експериментальних тварин (2002 р.), Конвенції Ради Європи про охорону хребетних тварин, що використовують в експериментах та інших наукових цілях від 18.03.1986 р., Директиві ЄС № 609 від 24.11.1986 р. і наказі МОЗ України № 281 від 01.11.2000 р., Законі України „Про захист тварин від жорстокого поводження” від 14.03.2006 р., етичному кодексі ученого України (2009р.), Біоетичній експертизі доклінічних та інших наукових досліджень, що виконуються на тваринах (Київ, 2006р.), Рекомендаціях Міжнародного Комітету Редакторів Медичних Журналів (ICMJE), Директиві Європейського Союзу 2010/10/63E4.

Голова комісії з біоетики  
Тернопільського національного медичного  
університету імені І. Я. Горбачевського  
МОЗ України, доктор медичних наук,  
професор, заслужений діяч науки і  
техніки України



С.Н. Вадзюк

Секретар комісії з біоетики



О.М. Ратинська

Особистий підпис

Заступник ректора з кадрових питань  
Тернопільського національного  
медичного університету

